

(19) World Intellectual Property Organization  
International Bureau(43) International Publication Date  
12 July 2001 (12.07.2001)

PCT

(10) International Publication Number  
**WO 01/50319 A2**(51) International Patent Classification: **G06F 17/00**(US) **KNAUFF, Christopher** [US/US]; 11813 Aspen View Drive, San Diego, CA 92128 (US).(21) International Application Number: **PCT/US00/35103**(74) **Agent: DELANEY, Karoline, A.** Knobbe, Martens, Olson & Bear, LLP, 620 Newport Center Drive, 16th Floor, Newport Beach, CA 92660 (US).

(22) International Filing Date:

22 December 2000 (22.12.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

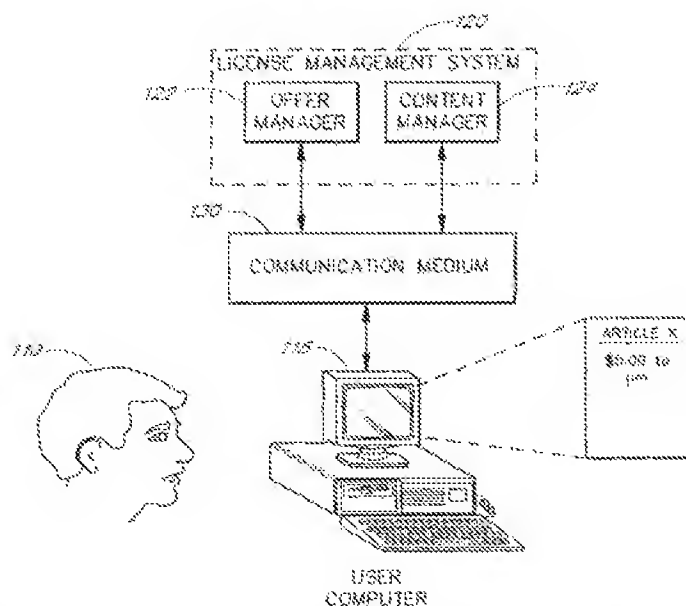
09/474,830 30 December 1999 (30.12.1999) US

(71) Applicant (for all designated States except US): **MEDIAADNA, INC.** [US/US]; 2223 Avenida de la Playa, Suite 105, La Jolla, CA 92037 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MY, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: **SYSTEM AND METHOD FOR PROVIDING ELECTRONIC LICENSES**

(57) **Abstract:** The present invention relates to a system and method for providing a license management system wherein customers utilize computers to connect to a license management system via a communication medium. Customers interact with the license management system via a communication medium to obtain copies of various pieces of content. In addition, the customers interact with the license management system via a communication medium to obtain licenses for accessing the content. The license management system can be used with varying types of content and allows the owner of the content to have full control of access to the content.

**Published:**

— Without international search report and to be republished upon receipt of that report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

## SYSTEM AND METHOD FOR PROVIDING ELECTRONIC LICENSES

### Background of the Invention

#### Field of the Invention

The present system and method relates to digital rights management.

#### 5 Description of the Related Technology

The growth of the Internet has created an increase in the number of on-line consumers as well as on-line companies that provide goods and services via the Internet. Accordingly, consumers have begun to rely more heavily upon the electronic transfer of information and expect to access information instantaneously. While such ease of access is of great benefit to the consumer, being able to control consumer access is often a difficult task.

10 One approach to controlling consumer access involves the use of digital rights management (DRM). DRM attempts to attach digital rights to information thereby controlling user access and affording the owner of the information adequate control. For example, DRM may allow the owner of the information to obtain royalty payments, track usage of the information, issue licenses, and so forth.

One common problem is that conventional approaches fail to adequately control consumer access to  
15 information. Consumers may find ways to get around the owner's control. For example, one owner may send the consumer a password to access the information. Once the consumer accesses the information, the consumer may make copies of the information and pass it along to others.

Another common problem is that conventional approaches fail to provide a digital rights management system that works for various types of information. For example, one digital rights management system may work for music  
20 files, but may not be appropriate for literary works.

### Summary of the Invention

The present invention provides a system and associated methods for granting access to a data object.

The present invention relates to a system and method for providing a license management system wherein customers utilize computers to connect to a license management system via a communication medium. Customers  
25 interact with the license management system via a communication medium to obtain copies of various pieces of content wherein a content cell containing the requested piece of content is created and sent to the customer. In addition, the customers interact with the license management system via a communication medium to obtain a license for accessing the content wherein a license cell containing the requested license is created and sent to the customer. Once the customer has both the content cell and a related license cell, the customer is allowed to access the piece of  
30 content as prescribed by the terms of the content cell and the license cell.

One advantage of the system and method is that the license management system provides complete control over a customer's access to information. The license management system can embed the license rights that are associated with the customer and the access privileges that are related to the content within the content and/or license. Thus, customers receive access to pieces of content after the license requirements have been satisfied.

Furthermore, license rights permit the owner to regulate various types of access to a piece of content. For example, a customer may be able to view a piece of content, but may be restricted from printing a copy of the piece of content.

Another advantage of the system and method is that the license management system works for a variety of different types of information. A content cell may be created for a wide variety of content and a license is created that  
5 corresponds to the content cell, thus allowing licenses to be created for any content cell. For example, a customer may obtain access to various types of content such as sound files, video files, executable files, text files, and so forth.

An additional benefit of this embodiment is that customers can receive a piece of content and access the content without much delay because the license management system is available for immediate access via the communication medium. The license management system ensures that the user receives timely responses to its  
10 requests.

For purposes of summarizing the invention, certain aspects, advantages, and novel features of the invention are described herein. It is to be understood that not necessarily all such advantages may be achieved in accordance with any particular embodiment of the invention. Thus, for example, those skilled in the art will recognize that the invention may be embodied or carried out in a manner that achieves one advantage or group of advantages as taught  
15 herein without necessarily achieving other advantages as may be taught or suggested herein.

#### Brief Description of the Drawings

Figure 1 illustrates a high level block diagram of one embodiment of the present invention and illustrates the interaction between the license management system and the user computer.

Figure 2 illustrates a high level block diagram of one embodiment of the present invention showing the flow  
20 of information among the content manager, the offer manager, and the user computer.

Figure 3 illustrates a block diagram of one embodiment of the user computer of Figure 2.

Figure 4 illustrates a block diagram of one embodiment of the content manager of Figure 2.

Figure 5 illustrates a block diagram of one embodiment of the offer manager of Figure 2.

Figure 6a illustrates a block diagram of one embodiment of a content cell.

Figure 6b illustrates a block diagram of another embodiment of a content cell.

Figure 7a illustrates a block diagram of one embodiment of a license cell.

Figure 7b illustrates a block diagram of another embodiment of a license cell.

Figure 8 illustrates a flowchart of one embodiment of providing a user with access to content.

Figure 9 illustrates a flowchart of one embodiment of sending content.

Figure 10 illustrates a flowchart of one embodiment of requesting a license.

Figure 11 illustrates a flowchart of one embodiment of sending a license.

Figure 12 illustrates a flowchart of one embodiment of checking for content and a license.

Figure 13 illustrates a flowchart of one embodiment of creating a content cell.

Figure 14 illustrates a flowchart of one embodiment of creating a license cell.

### Detailed Description

These and other features will now be described with reference to the drawings. These drawings and the associated description are provided to illustrate embodiments of the invention, and not to limit the scope of the invention.

Throughout the drawings, reference numbers are re-used to indicate correspondence between referenced elements. In addition, the first digit of each reference number indicates the figure in which the element first appears.

This application incorporated by reference U.S. Patent No. 5,845,281 to Benson, et al.

The present invention relates to a system and method for providing a license management system 120. Figure 1 illustrates one embodiment in which customers 110 utilize user computers 115 to connect to a license management system 120 via a communication medium 130. In the exemplary system, the license management system 120 comprises two primary components, an offer manager 122 and a content manager 124. Customers 110 interact with the content manager 124 via the communication medium 130 to obtain copies of various pieces of content. Pieces of content may include sound files, video files, text documents, e-mail, database records, HyperText Markup Language (HTML) files, Extensible Markup Language (XML) files, Electronic Data Interchange (EDI) files, Message Oriented Middleware (MOM) files, executable scripts, and so forth. In addition, customers 110 interact with the offer manager 122 via the communication medium 130 to obtain licenses for accessing the pieces of content.

#### I. Overview

An overview of one embodiment of a license management system 120 is shown in Figure 2. In the exemplary system, a user computer 115 communicates with the offer manager 122 and the content manager 124, via a communication medium 130. In the embodiment depicted in Figure 2, the offer manager 122 and the content manager 124 of the license management system 120 are implemented as separate components connected to a communication medium 130. It is recognized that in other embodiments, the offer manager 122 and the content manager 124 may be part of a single system such that they may be directly connected.

Figure 2 also illustrates flow of information when a customer 110 (Figure 1) wants to access a piece of content. With reference to event A, the user computer 115 requests a piece of content and the request is sent to the content manager 124. In event B, the content manager 124 sends a content cell 210 containing the requested content to the user computer 115. Next, in event C, the user computer 115 requests a license and the request is sent to the offer manager 122. In event D, the offer manager 122 sends a license cell 220 containing the requested license to the user computer 115. In events E and F, the user computer 115 verifies that it has both the content cell 210 and a corresponding license cell 220, and the user computer 115 allows access to the content.

One benefit of this embodiment is that the user computer 115 can receive a piece of content and access the content without much delay because the license management system 120 is available for immediate access via the

communication medium 130. The license management system 120 ensures that the user receives timely responses to its requests.

Another benefit of this embodiment is that the license management system 120 controls the user computer's 115 use of the content via the license. The license management system 120 can embed the license rights that are associated with the user computer 115 and the access privileges that are related to the content within the content cell 210 and/or within the license cell 220.

## II. Implementation of a License Management System

This section provides further description of one embodiment of a system for providing electronic licenses as shown in Figure 2. The exemplary system includes a communication medium 130, a user computer 115, and a license management system 120.

### A. Communication Medium

Focusing now on the communication medium 130 as shown in Figure 2, the presently preferred communication medium 130 includes the Internet which is a global network of computers. The structure of the Internet, which is well known to those of ordinary skill in the art, includes a network backbone with networks branching from the backbone. These branches, in turn, have networks branching from them, and so on. Routers move information packets between network levels, and then from network to network, until the packet reaches the neighborhood of its destination. From the destination, the destination network's host directs the information packet to the appropriate terminal, or node. For a more detailed description of the structure and operation of the Internet, please refer to "The Internet Complete Reference," by Harley Hahn and Rick Stout, published by McGraw-Hill, 1994.

In one advantageous embodiment, the Internet routing hubs comprise domain name system (DNS) servers, as is well known in the art. DNS is a Transfer Control Protocol/Internet protocol (TCP/IP) service that is called upon to translate domain names to and from Internet Protocol (IP) addresses. The routing hubs connect to one or more other routing hubs via high speed communication links.

One of ordinary skill in the art, however, will recognize that a wide range of interactive communication mediums 130 may be employed in the present invention. For example, the communication medium 130 may include interactive television networks, telephone networks, wireless data transmission systems, two-way cable systems, customized computer networks, interactive kiosk networks, automatic teller machine networks, and the like.

One popular part of the Internet is the World Wide Web. The World Wide Web contains different computers which store documents capable of displaying graphical and textual information. The computers which provide information on the World Wide Web are typically called "websites." A website is defined by an Internet address which has an associated electronic page. The electronic page can be identified by a Uniform Resource Locator (URL). Generally, an electronic page is a document which organizes the presentation of text, graphical images, audio, video, and so forth.

### B. The User Computer

The user computer 115, shown in Figure 3, is a device which allows a user to interact with the communication medium 130 (Figure 1). In one embodiment, the user computer 115 is a conventional general purpose computer using one or more microprocessors, such as, for example, a Pentium processor, a Pentium II processor, a Pentium Pro processor, an x86 processor, an 8051 processor, a MIPS processor, a Power PC processor, or an Alpha processor. In one embodiment, the user computer 115 runs an appropriate operating system, such as, for example, Microsoft® Windows® 3.X, Microsoft® Windows 98, Microsoft® Windows® NT, Microsoft® Windows® CE, Palm Pilot OS, Apple® MacOS®, Disk Operating System (DOS), UNIX, Linux®, or IBM® OS/2® operating systems. In one embodiment, the user computer 115 is equipped with a conventional modem or other network connectivity such as, for example, Ethernet (IEEE 802.3), Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI), or Asynchronous Transfer Mode (ATM). As is conventional, in one embodiment, the operating system includes a TCP/IP stack which handles all incoming and outgoing message traffic passed over the communication medium 130.

In other embodiments, the user computer 115 may, for example, be a computer workstation, a local area network of individual computers, an interactive television, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, a telephone, a router, a satellite, a smart card, an embedded computing device, or the like which can interact with the communication medium 130. While in such systems, the operating systems will differ, they will continue to provide the appropriate communications protocols needed to establish communication links with the communication medium 130.

The exemplary user computer 115 includes a browser module 310, a user manager 320, as well as a database collection 330.

#### 1. Browser Module

In one embodiment, the user computer 115 utilizes several operational modules including a user browser module 310. The user browser module 310 (hereinafter referred to as the user's browser 310) is a software program which allows a consumer to access different content providers through the communication medium 130 (Figure 1). In one embodiment, the user's browser 310 is the Netscape® Navigator developed by Netscape, Inc. or the Microsoft® Internet Explorer developed by Microsoft Corporation. One of ordinary skill in the art, however, will recognize that numerous other types of access software could also be used to implement an embodiment of the present invention. These other types of access software could be, for example, other types of Internet browsers, custom network browsers, two-way communications software, cable modem software, point-to-point software, and the like.

#### 2. The User Manager

In one embodiment, the user computer 115 includes a user manager 320. The user manager 320 tracks and manages all content and licenses received by the user computer 115. In one embodiment, the user manager 320 may be loaded onto the user computer 115 through a web-based download process. In the web-based download process, the user's browser 310 may ask the user if the user would like to download the user manager 320. If the user responds yes, then the download process checks to see if the user manager 320 supports the user's browser 310. If so, then the user's

browser 310 automatically downloads the user manager 320 onto the user's computer. It is recognized that in other embodiments, the user manager download process could be performed without any user interaction or the user manager 320 could be pre-installed in the user computer 115.

5

### 3. The Database Collection

In one embodiment, the user computer 115 includes a database collection 330 shown in Figure 3. The exemplary database collection 330 includes several databases such as a license database 332, a content database 334, and an encryption key database 336, which may contain one or more cryptographic keys.

10 The license database 332 contains information about license cells 220 that have been received by the user computer 115. The information may include a copy of the license cells 220 as they are received, or extracted information such as the license rules, the license cell header information, and so forth.

The content database 334 contains information about content cells 210 that have been received by the user computer 115. The information may include a copy of the content cells 210 as they are received, or extracted  
15 information such as the encrypted content, the content cell header information, and so forth.

The encryption key database 336 includes the keys received via the license cell 220 that the user manager 320 may use to decrypt the encrypted content. While one embodiment uses symmetric key encryption, it is recognized that in other embodiments, other methods of encryption may be used separate from or in conjunction with symmetric encryption.

20 In one embodiment, the database collection 330 is implemented using a flat file database structure. It is recognized that in other embodiments, the database collection 330 could be implemented using different databases such as the relational database Microsoft® SQL Server allowing access to the data via the Structure Query Language (SQL). Moreover, while the database collection 330 depicted in Figure 3 is comprised of several separate databases, it is recognized that in other embodiments, the database collection 330 may contain other databases or some of the  
25 databases could be combined. In addition, the database collection 330 may be implemented as a single database with separate tables or as other data structures that are well known in the art such as linked lists, binary trees, and so forth.

### C. The License Management System

In one embodiment, the license management system 120 (Figure 1) manages the creation and issuance of content and licenses. It is recognized that in other embodiments, the license management system 120 may primarily  
30 manage the issuance of content cells 210 and license cells 220; and other modules could create the license cells 220 and/or the content cells 210. As previously discussed, the license management system 120 comprises two primary components, the content manager 124 and the offer manager 122. It is recognized that in other embodiments, however, there could be multiple content managers 124 and/or multiple offer managers 122.



## 1. The Content Manager

Figure 4 illustrates one embodiment of a content manager 124. The exemplary content manager 124 includes web server software 410, a content manager component 420, as well as a database collection 430.

### a. Web Server Software

5 In one embodiment, the content manager 124 includes web server software 410 as shown in Figure 4. The web server software 410 may be, for example, Netscape's Internet Server software, Microsoft's Internet Server Software (ISS), or the like. Such web server software 410 is configured to process messages from the user computer 115 and the offer manager 122 as illustrated in Figure 2.

### b. Content Manager Component

10 In one embodiment, the content manager 124 includes a content manager component 420 that manages the creation and issuance of content cells 210. The content manager component 420 illustrated in Figure 4 comprises two processes, the user manager download process 422 and the content cell creation process 424. The content manager component 420 may also include other processes not shown such as managing user registration, tracking the inventory of the content cells, as well as providing a search engine.

15 The user manager download process 422 downloads a copy of the user manager 320 onto the user computer 115. It is recognized that in other embodiments the user manager download process 422 could be performed by other modules or the user manager 320 could be pre-installed on the user computer 115. The content cell creation process 424 may be invoked when the user computer 115 requests a piece of content, and the process 424 creates a content cell 210 with the requested content. While in this embodiment the content manager 124 includes a contents creation process 424, it is recognized that in other embodiments, other modules may create the content cell 210, and the content manager 124 may primarily manage the issuance of the content cell 210 by locating the content cell 210 rather than creating the content cell 210.

### c. Database Collection

25 In one embodiment, the content manager 124 includes a database collection 430 shown in Figure 4. The exemplary database collection 430 includes several databases such as a rules database 432, an encryption key database 434, which may contain one or more symmetric keys, a public certificate database 436, a content database 438, and a cell header information database 439.

30 The rules database 432 contains rules pertaining to how and when to assign rights to a piece of content. The rules can be general (e.g., Content X requires License Y) or they can be more specific (e.g., Anyone with License Y can print Content X; To print Content X, need License Y, also go to URL C). While Figure 4 shows the rules as being stored in a rules database 432, it is recognized that in other embodiments, the rules may be stored in other formats such as a program or a set of scripts.

The encryption key database 434 contains a set of symmetric keys that the content manager 124 may use to encrypt the requested content. The symmetric keys may be created using techniques well known in the art. For a

more detailed description of symmetric and asymmetric encryption, please refer to "Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition," by Bruce Schneier, published by John Wiley & Sons, 1996. While this embodiment uses symmetric encryption, it is recognized that in other embodiments, other methods of encryption may be used separate from or in conjunction with symmetric encryption.

5           The public certificate database 436 contains public certificates from other components with which the content manager 124 communicates. A component's public encryption key resides within the component's public certificate. For example, the public certificate database 436 may contain the public certificates of the offer manager 122, which contain the public keys of the offer manager 122, to allow for encryption and authentication when the content manager 124 and the offer manager 122 communicate. The public certificate database 436 may also contain  
10   public certificates of the user computers 115 that request pieces of content.

          The content database 438 contains information about various pieces of content as well as the content itself. Pieces of content may include sound files, video files, text documents, e-mail, database records, HyperText Markup Language (HTML) files, Extensible Markup Language (XML) files, Electronic Data Interchange (EDI) files, Message  
15   Oriented Middleware (MOM) files, executable scripts, and so forth. The content database 438 may include tables such as a publishers table, a publications table, and an articles table.

          The publishers table tracks information about the publishers. The publishers table may include fields such as publisher ID, status (signifying whether the publisher is active or inactive), publisher name, default price, and so forth. Each publisher may publish one or more publications.

          The publications table tracks information about the publications. The publications table may include fields  
20   such as publication ID, status (signifying whether the publication is active or inactive), publication code, properties, publisher ID, domain, publication name, publication file type, publication web address, introductory credit, default price, number of free articles, print field (signifying whether printing is permitted), copy field (signifying whether copying is permitted), and so forth. Each publication may include one or more articles.

          The articles table tracks information about the articles, or pieces of content. While the term "article" is used,  
25   it is recognized that the term "article" may include pieces of content such as sound files, video files, text documents, e-mail, database records, HyperText Markup Language (HTML) files, Extensible Markup Language (XML) files, Electronic Data Interchange (EDI) files, Message Oriented Middleware (MOM) files, executable scripts, and so forth. The articles table may include fields such as title, time stamp, publication ID, publisher ID, price, valid time period, cell type, and so forth.

30           The content database 438 may include other tables that relate to the content such as a subscriptions table, an administrators table, and so forth. The subscriptions table defines various subscriptions and includes fields such as a description field, publication ID, price, and so forth. The administrators table tracks special accounts that are authorized to perform administrative tasks such as setting prices, expiration dates, groups, and so forth. The

administrators table may include fields such as name, password, email address, telephone number, publisher ID, administrator level, permission to manage group accounts, permission to manage promotion, and so forth.

The call header information database 439 contains information about the call header. This information may include unique identifiers for each content cell 210, information about various stores that produce the pieces of content, descriptive information, as well as validation information (e.g., expiration date/time).

In connection with the database collection 430, in one embodiment, there may be several processes (not shown) such as ID generators, number generators, statistic generators, session generators and temp storage units that work with the database collection 430.

In one embodiment, the database collection 430 is implemented using the relational database Microsoft® SQL Server allowing access to the data via the Structured Query Language (SQL). SQL is a language standardized by the International Standards Organization for defining, updating, and querying a relational database.

It is recognized that in other embodiments, the database collection 430 could be implemented using a different relational database as well as other types of databases (such as a flat file database). Moreover, while the database collection 430 depicted in Figure 4 is comprised of several separate databases, it is recognized that in other embodiments, the database collection 430 may contain other databases or some of the databases could be combined. In addition, the database collection 430 may be implemented as a single database with separate tables or as other data structures that are well known in the art such as linked lists, binary trees, and so forth.

In one embodiment, the database collection 430 may be connected to a backend component (not shown) that receives database requests via servlets, small programs that run on servers, and sends a corresponding SQL request to the database collection 430. It is recognized that in other embodiments data access could be performed differently, for example, a different type of backend component could be used or the database collection 430 could be accessed directly.

A more detailed description of one embodiment of a database collection 430 can be found in the attached Appendix A. While the database collection 430 in Appendix A illustrates a license management system 120 wherein the content manager 124 and the offer manager 122 both have access to the same database collection, it is recognized that in other embodiments, the content manager 124 and the offer manager 122 may have separate copies of various databases.

## 2. The Offer Manager

Figure 5 illustrates one embodiment of an offer manager 122. The exemplary offer manager 122 includes web server software 510, an offer manager component 520, a license manager 530, and a database collection 540.

### a. Web Server Software

In one embodiment, the offer manager 122 includes web server software 510 as shown in Figure 5. The web server software 510 may be, for example, Netscape's Internet Server software, Microsoft's Internet Server software

(ISS), or the like. Such web server software 510 is configured to process messages from the user computer 115 and the content manager 124.

h. Offer Manager Component

In one embodiment, shown in Figure 5, the offer manager 122 includes an offer manager component 520 that manages the issuance of license cells 220. The offer manager component 520 may also include processes (not shown) such as managing interaction with users, determining whether user requirements have been satisfied before the license manager 530 is invoked, such as tracking monetary payments, determining whether a license was issued, as well other processes.

10

c. License Manager Component

In one embodiment, the offer manager 122 includes a license manager 530 that manages the creation of license cells 220. The license manager 530 illustrated in Figure 5 is comprised of a license cell creation process 532. The license cell creation process 532 may be invoked after the user computer 115 requests a license for a content cell 210. The license cell creation process 532 creates a license cell 220 for the requested content. While in this embodiment the license manager 530 includes a contents creation process 424, it is recognized that in other embodiments, other modules may create the license cell 220.

For example, the offer manager 122 could include a license cell creation process 532 and create the license cells 220. However, in one embodiment, another module may create the license cell 220 such that either the license manager 530 or the content manager 124 need only locate the appropriate license cell 220.

The license manager 530 may also include other processes not shown such as managing the repository of issued licenses. In addition, in one embodiment, the license manager 530 accesses the issued license database 547 tracking each time a new license is created. This management role allows the license manager 530 to act as an audit trail of the licensing process and to provide requested license information to other modules.

25

d. Database Collection

In one embodiment, the offer manager 122 includes a database collection 540 shown in Figure 5. The exemplary database collection 540 includes several databases such as a rules database 541, an encryption key database 542, which may contain one or more symmetric keys, a public certificate database 543, a content database 544, a user database 545, a group database 546, an issued license database ("license database") 547, a billing database 548, and a cell header information database 549.

The rules database 541 contains rules about how and when to assign rights to a piece of content. The rules can be general (e.g., Content X requires License Y) or they can be more specific (e.g., Anyone with License Y can print Content X; To print Content X, need License Y, also go to URL C). While Figure 5 depicts a rules database 541, it is recognized that in other embodiments, the rules may be stored in other formats such as a program or a set of scripts.

The encryption key database 542 contains a set of symmetric keys used by the content manager 124 to encrypt the pieces of requested content. The symmetric keys may be created using techniques well known in the art. For a more detailed description of symmetric and asymmetric encryption, please refer to "Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition," by Bruce Schneier, published by John Wiley & Sons, 1996. While this embodiment uses symmetric encryption, it is recognized that in other embodiments, other methods of encryption may be used separate from or in conjunction with symmetric encryption.

The public certificate database 543 contains a set of public certificates from other components with which the offer manager 122 communicates. A component's public encryption key resides within the component's public certificate. For example, the public certificate database 543 may contain the public certificates of the content manager 124, which contain the public keys of the content manager 124, to allow for authentication when the content manager 124 and the offer manager 122 communicate. The public certificate database 543 may also contain public certificates of the user computers 115 that request licenses.

In certain embodiments, the offer manager's content database 544 may include part or all of the content manager's content database 438. The content database 544 may copy data from the content manager's content database 438 using data replication techniques well known in the art allowing the offer manager 122 and the content manager 124 to each have a copy of the same data without destroying the data's integrity. Moreover, the offer manager's content database 544 may include information not in the content manager's content database 438. While this embodiment depicts the offer manager 124 and the content manager 122 each having its own copy of the same content database 544, 438, it is recognized that in other embodiments, the offer manager 122 and the content manager 124 may both have access to the same content database. For detailed information on one embodiment of the content database 544, please refer to section entitled 1. Content Manager - c. The Database Collection.

The user database 545 tracks information on the users. The user database 545 may contain tables such as a user information table, a user credit card information table, a user comments table, and a user account balances table.

The user information table includes information about the users of the system. The user information table may include fields such as user account ID, first name, last name, company, e-mail address, account creation date, account status, current charges, current credits, outstanding limit, service level, promotion e-mail, remote user, remote address, remote host, billing name, billing address, password, password encryption type, and so forth. The user credit card information table keeps a record of the user's encrypted credit card number as well as other credit card information. The user credit card information table may include fields such as user account ID, credit card encryption method, encrypted credit card account number, credit card expiration month, credit card expiration year, and so forth. The users comments table tracks comments specific to a particular user, allowing for customized customer support. The user comments table may include fields such as user account ID, time stamp, comment field, and so forth. The user account balance table keeps a record of a user's credit and charges for each specific publication a user has

accessed. The user account balance table may include fields such as user account ID, publication ID, time credit was created, available credit, credit amount applied, current charge, pricing terms displayed, as well as other information related to the user's account.

The group database 546 allows users to be consolidated into a single account, such as a corporate account.

5 The group database 546 may include tables such as a groups table, a group users table, and a publications table.

The groups table maintains group-related information. The groups table may include fields such as group ID, group name, password, type of account, number of days the accounts are valid, publication ID, expiration date, and so forth. The group users table identifies the members of a group. The group users table may include fields such as group ID, user account ID, expiration date, and so forth. The group publications table identifies which publications can  
10 be accessed by the group's members. The group publications table may include fields such as group ID, publication ID, and so forth.

The license database 547 may contain several tables such as an issued license table. The issued license table records every license issued to a user that allows access to an article. The issued license table may include fields such as transaction ID, time stamp, article ID, content cell location, content cell ID, framework ID, publication  
15 ID, amount charged, and so forth.

The billing database 548 manages the billing transactions. The billing database 548 may include tables such as current transactions, billable transactions, billable summaries, and so forth. It is recognized that other standard billing information may included in the billing database 548.

The cell header information database 549 contains information for the cell header. This information may  
20 include unique identifiers for each license cell 220, unique identifiers for each content cell 210, information about various stores that produce the pieces of content, descriptive information, as well as validation information (e.g., expiration date/time).

The database collection 540 may also include other databases (not shown) for performing various management tasks. For example, the database collection 540 may include a user action database that tracks various  
25 user activity such as user requests, use click throughs (tracking each time a user selects a web link), and so forth. In addition, the database collection 540 may include an error database that tracks various system errors such as errors related to specific articles, errors related to new users, errors related to new licenses, and so forth. The database collection 540 may also include a database for maintaining data integrity, such as various synchronization tables that alert the system when new information should be added and for existing information should be changed. The database  
30 collection 540 may also include a promotions database that tracks the promotions sent to the user via email and other communication means. The database collection 540 may also include user manager activation information for tracking the number of times the user attempts to load the user manager 320 onto one the of the user's computing devices, when the user attempts to reinstall the user manager 320 onto one of the user's computing devices, when the user has verified the user's ID, and so forth.

In connection with the database collection 540, in one embodiment, there may be several processes (not shown) such as ID generators, number generators, statistic generators, session generators and temp storage units that work with the database collection 540.

In one embodiment, the database collection 540 is implemented using the relational database Microsoft®  
 5 SQL Server allowing access to the data via the Structured Query Language (SQL). SQL is a language standardized by the International Standards Organization for defining, updating, and querying a relational database.

It is recognized that in other embodiments, the database collection 540 could be implemented using a different relational databases as well as other types of databases (such as a flat file database). Moreover, while the database collection 540 depicted in Figure 5 is comprised of several separate databases, it is recognized that in other  
 10 embodiments, the database collection 540 may contain other databases or some of the databases could be combined. In addition, the database collection 540 may be implemented as a single database with separate tables or as other data structures that are well known in the art such as linked lists, binary trees, and so forth.

In one embodiment, the database collection 540 may be connected to a backend component (not shown) that receives database requests via servlets, small programs that run on servers, and sends a corresponding SQL request to  
 15 the database collection 540. It is recognized that in other embodiments data access could be performed differently, for example, a different type of backend component could be used or the database collection 540 could be accessed directly.

A more detailed description of one embodiment of a database collection 540 can be found in the attached Appendix A. While the database collection 540 in Appendix A illustrates a license management system 120 wherein  
 20 the content manager 124 and the offer manager 122 both have access to the same database collection, it is recognized that in other embodiments, the content manager 124 and the offer manager 122 may have separate databases, which may be alike, partly similar or different.

### III. License Management System Cells

In one embodiment, the license management system 120 creates cells that contain content (content cells  
 25 210) as well as cells that contain licenses (license cells 220).

#### A. Content Cell

Figure 6a illustrates one embodiment of a content cell 210a. The exemplary content cell 210a contains a set of rules 612a, encrypted content 614a, a cell header 616a, a public certificate 618a, as well as a content cell  
 signature 611a.

#### 30 1. The Rules

The content cell rules 612a tell the user computer 115 what rights are associated with the content cell 210a and what is required in order to have those rights. For example, a content cell 210a may have one set of requirements for the right to print an article, and another set of requirements for viewing an article. In one embodiment, the rules



612a may be written for a specific user, while in other embodiments, the rules 612a may be written more generically and may apply to a particular class of users.

## 2. Encrypted Content

The content cell's encrypted content 614a contains the requested content encrypted by the content manager  
5 124. In one embodiment, the content is encrypted using a symmetric key. The symmetric key is then encrypted by the license management system 120 using the offer manager's public key so that only the offer manager 122 can decrypt the symmetric key and the encrypted symmetric key is sent to the user manager 320. When the user manager 320 requests a license, the user manager 320 sends the offer manager 122 the encrypted symmetric key. Once the offer manager 122 verifies that the user computer 115 has a valid license, the offer manager 122 can decrypt the  
10 symmetric key and encrypt the symmetric key with the user manager's public key allowing the user manager 320 to decrypt the symmetric key and thus decrypt the content. For a more detailed description of symmetric and asymmetric encryption, please refer to "Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition," by Bruce Schneier, published by John Wiley & Sons, 1996. While this embodiment describes a use of a combination of symmetric encryption and asymmetric encryption, it is recognized that other methods of encryption may be used. For  
15 example, the offer manager 122 and the content manager 124 could both have access to a database of symmetric keys such that the symmetric key is not sent to the user computer 115 until the user computer 115 has obtained a license.

## 3. Cell Header

The cell header 618a contains information about the content cell 210a. Such information may include a  
20 unique identifier for the content cell 210a, information about the store that produced the requested content, descriptive information, information for locating an offer manager 122 (e.g., the URL of the offer manager), as well as validation information (e.g., expiration date/time).

## 4. Public Certificate

The content cell's public certificate 618a is the public certificate of the cell creator which contains the cell  
25 creator's public key. For example, if the content manager 124 creates the content cell 210a, then the content manager 124 would sign the cell 210a and include its public certificate 618a in the content cell 210a. The public certificate 618a allows the recipient of the cell (e.g., the user computer 115) to verify the authenticity of the sender (e.g., the content manager 124).

## 5. Content Cell Signature

30 The content cell's digital signature 611a is a piece of information based on both the content cell 210a and the content manager's private keys. The digital signature 611a can be used by the user computer 115 to authenticate the creator of the content cell 210a and the information contained in the content cell 210a.

In general, a digital signature is a code that is unique for each message and to each message sender. Authentication using digital signatures assures the receiver that the message was from the expected sender, and it



confirms that the message is complete and has not been altered. The digital signature is a cryptographic means through which the origin of the cell and the identity of the sender may be verified. For a more detailed description of digital signatures, please refer to "Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition," by Bruce Schneier, published by John Wiley & Sons, 1996.

5. Other Signatures

Figure 5b illustrates another embodiment of a content cell 210b. While the rules 612b, the encrypted content 614b, and the cell header 616b are similar to those described above, each component has its own digital signature 613b, 615b, 617b. Each digital signature 613b, 615b, 617b is created by the module that created the component. For example, the rules signature 613b is created by the module that created the rules 612b; the cell header signature 617b is created by the module that created the cell header 616b. In other embodiments, different modules may create different parts of the content cell 210, and each module provides its own digital signature. This functionality permits the recipient of the cell to verify the identity of the component creator. In order to verify the creator's identity in this embodiment, the content cell 210 must contain a public certificate 618b for each module that created a part of the content cell 210.

15 In other embodiments, digital signatures could also be used for individual rules or sets of rules. For example, if Module A creates Rules X, Y, and Z and Module B creates Rules M and N, then Module A could sign Rules X, Y, and Z, and Module B could sign Rules M and N. The public certificates of both Module A and Module B would be sent in the content cell 210 so the user could verify the origin of each rule.

It is recognized that in other embodiments, the content cell 210 may be comprised of different components or  
20 may include only a subset of the above mentioned components. Furthermore, while various encryption techniques such  
as symmetric encryption, asymmetric encryption, and digital signatures have been discussed, it is recognized that in  
other embodiments other means of ensuring security may be used.

8. License Cell

Figure 7a illustrates one embodiment of a license cell 220a. The exemplary license cell 220a contains a set of encrypted rules 712a, an encrypted license 714a, a cell header 716a, a public certificate 718a, as well as a license cell signature 711a.

## 1. The Rules

The license cell rules 712a tell the user computer 115 what rights are associated with the license cell 220a and what is required in order to have those rights. For example, a license cell 220a may have one set of requirements for the right to print an article and another set of requirements for viewing an article. In one embodiment, the rules may be written for a specific user, while in other embodiments, the rules may be written more generically and may apply to a particular class of users. In one embodiment, both the license cell 220 and the content cell 210 contain rules 612, 712. In other embodiments, only the license cell 220, only the content cell 210, or neither contain rules 612, 712.

## 2. Encrypted License

The license cell's encrypted license 714a contains the requested license encrypted by the license manager 530. In one embodiment, the license is a unique identifier. It is recognized that in other embodiments, the license could be another type of data such as a record of information and/or the license could include the rules.

5 In one embodiment, the license is encrypted using a symmetric key. For example, the license could be encrypted with the same symmetric key used to encrypt the content. In one embodiment, the symmetric key is then encrypted by the license management system 120 using the user manager's public key so that only the user manager 320 can decrypt the symmetric key and thus decrypt the license. While this embodiment describes a use of a combination of symmetric encryption and asymmetric encryption, it is recognized that other methods of encryption  
10 may be used.

## 3. Cell Header

The cell header 716a contains information about the license cell 220a. Such information may include a unique identifier for the license cell 220a, an identifier for related content cells 210, information about the store that produced the requested content, descriptive information, as well as validation information (e.g., expiration date/time).

## 4. Public Certificate

15 The license cell's public certificate 718a is the public certificate of the cell creator which contains the cell creator's public key. For example, if the license manager 530 creates the license cell 220a, then the license manager 530 would sign the cell 220a and include its public certificate 718a in the license cell 220. The public certificate 718a allows the recipient of the cell (e.g., the user computer 115) to verify the authenticity of the sender (e.g., the license  
20 manager 530).

## 5. License Cell Signature

The license cell's digital signature 611a is a piece of information based on both the license cell 220a and the license manager's private keys. The digital signature 611a can be used by the user computer 115 to authenticate the creator of the license cell 220a and the information contained in the license cell 220a.

## 6. Other Signatures

25 Figure 7b illustrates another embodiment of a license cell 220b. While the rules 712b, encrypted license 714b, and cell header 716b are similar to those described above, each component has its own digital signature 713b, 715b, 717b. Each digital signature 713b, 715b, 717b is created by the module that created the component 712b, 715b, 716b. For example, the rules signature 713b is created by the module that created the rules 712b; the cell header signature 715b is created by the module that created the cell header 716b. In other embodiments, different  
30 modules may create different parts of the license cell 220b, and thus each module provides its own digital signature. This functionality permits the recipient of the cell to verify the identity of the component creator. In order to verify the creator's identity in this embodiment, the license cell 220b must contain a public certificate 718b for each module that created a part of the license cell 220b.

In other embodiments, digital signatures could also be used to sign individual rules or sets of rules. For example, if Module A creates Rules X, Y, and Z and Module B creates Rules M and N, then Module A could sign Rules X, Y, and Z, and Module B could sign Rules M and N. The public certificates of both Module A and Module B would need to be sent in the license call 220 so the user could verify the origin of each rule.

5 It is recognized that in other embodiments, the license call 220 may be comprised of different components or may include only a subset of the above mentioned components. Furthermore, while various encryption techniques such as symmetric encryption, asymmetric encryption, and digital signatures have been discussed, it is recognized that in other embodiments other means of ensuring security may be used.

#### IV. License Management Process Overview

10 Figure 8 illustrates a flowchart of one embodiment of the flow of information previously discussed in Figure 2. The flowchart illustrated in Figure 8 corresponds with the flow of information in Figure 2, wherein state 820 corresponds with event A; state 830 corresponds with event B; state 840 corresponds with event C; state 850 corresponds with event D; state 860 corresponds with event E; and state 870 corresponds with event F.

Beginning at the start state 810, the process proceeds to the user computer request state 820. In state 820, the user computer 115 requests a specific piece of content and then proceeds to the content manager send state 830. In state 830, the content manager 124 sends the requested piece of content to the user computer 115 and proceeds to the next state 840 in which the user computer 115 requests a license. In state 850, the offer manager 122 sends a license to the user computer 115 and proceeds to state 860 in which the user computer 115 checks for the requested content and the corresponding license. Once those items are found, the process proceeds to state 870  
20 wherein the user manager 320 invokes the appropriate application with the requested content. The process then proceeds to the end state 880.

##### A. User Requests Content

In one embodiment, in state 820, the user views a web page using the user's browser 310. Next, the user requests a piece of content to view, print, copy, play, etc. It is recognized that in other embodiments, the user could request a piece of content by attempting to access a content cell 210 sent to the user via e-mail, located on a storage unit such as a CD-ROM, floppy disc, and so forth. In addition, in other embodiments, the content request could occur without informing the user. For example, when a certain computer application is executed, the computer program application could automatically send a request to the content manager 124 requesting a piece of content.

##### B. Content Manager Sends Content Cell

30 With reference to state 830, the content manager 124 finds the requested piece of content and then prepares the content for presentation to the user computer 115. The flowchart corresponding to state 830 is further illustrated in Figure 9. Beginning at the start state 830, the process proceeds to state 910 wherein the content manager 124 receives the content request. Proceeding to state 920, the content manager 124 builds the content cell

210 containing the requested content and moves to state 930 wherein the content manager 124 sends the content cell 210 to the user's browser 310 on the user computer 115.

While in this embodiment the content manager 124 builds the content cells 210, it is recognized that in other embodiments the content manager 124 may locate a pre-built content cell 210 within a database. As previously described, other modules could build generic content cells 210 and place them in a repository such that another module, such as the content manager 124, could access the content cells 210 and send them to the user as requested. Additionally, the content cell 210 could be retrieved from a disk, a CD-ROM, as well as other storage devices. After the content manager 124 builds the content cell 210 in state 920, the content manager 124 sends the content cell 210 to the user's browser 310 in state 930 and then proceeds to the end state 940. It is recognized in other embodiments, the content cell 210 could be sent directly to the user manager 320 and not to the user's browser 310.

#### C. User Computer Requests A License

The flowchart corresponding to state 840 when the user computer 115 requests a license is illustrated in Figure 10. Beginning at the start state 840, the process proceeds to state 1010 wherein the user's browser 310 receives the content cell 210 and passes the content cell 210 to the user manager 320. The user manager 320 opens the content cell 210 in state 1020 and proceeds to state 1030 in which the user manager 320 verifies authenticity of the content cell 210. As previously described, the content cell 210, in one embodiment, contains a signature and a public certificate such that the user manager 320 can use the public certificate to authenticate the sender and the creator of the content cell 210. Proceeding to state 1040, the user manager 320 attempts to locate a license for the content cell 210. If a license is located in state 1050, then the process proceeds to end state 1090 and moves to state 870 wherein the user manager 320 invokes the application with the content. If, however, the license is not located, then the process proceeds to state 1070 wherein the user manager 320 stores the content cell information in the content database 334 and proceeds to state 1080 in which the user computer 115 requests a license. In one embodiment, the user manager 320 directs the user's browser 310 to send a license request to the content manager 124 and proceeds to the end state 1090.

#### D. Offer Manager Sends License

The flowchart corresponding to state 850 wherein the offer manager 122 sends the license to the user computer 115 is illustrated in Figure 11. Beginning at the start state 850, the process then proceeds to state 1110 wherein the offer manager 122 receives a license request and proceeds to state 1120 wherein the offer manager 122 has the user complete license issuance requirements via the user's browser 310. For example, the user may have to submit registration information, mailing information, fill out a questionnaire, submit credit card information, and so forth. A sample registration form may include requests for information such as: first name, last name, e-mail address, company, credit card type, credit card number, expiration date, name on credit card, billing address, city, state, zip code, country, password, and so forth. Proceeding to state 1130, if the user does not complete the requirements, then the process proceeds to the end state 1140. If, however, the user does complete the requirements, then the process

proceeds to state 1150 wherein the user manager 320 begins to build the license cell 220. After the license manager 530 builds the license cell 220, the process proceeds to state 1160 wherein the offer manager 122 may "charge" the user computer 115 for the license. This charging takes place, for example, if the license issuance requirement included some type of monetary payment via credit card or other account information. It is recognized that in other  
5       embodiments, the billing transactions may be processed by another module such as a service provider, a server, and so forth. Proceeding to state 1170, the offer manager 122 then sends the license cell 220 to the user's browser 310 and proceeds to the end state 1180.

#### E.       User Computer Locates License And Content

The flowchart corresponding to state 860 wherein the user computer 115 locates the content and license is  
10       illustrated in Figure 12. Beginning in the start state 860, the process proceeds to state 1210 wherein the user's browser 310 receives the license cell 220 and passes the license cell 220 to the user manager 320. In state 1220, the user manager 320 opens the license cell 220 and then proceeds to state 1230 wherein the user manager 320 verifies the authenticity of the license cell 220. As previously described, the license cell 220, in one embodiment, contains a signature and public certificate such that the user manager 320 can use the license cell 220 to authenticate the sender  
15       and creator of the license cell 220. Proceeding to state 1240, the user manager 320 stores the license cell information in the user computer 115. In one embodiment, the license manager 530 stores the license cell information in the license database 332. It is recognized that in other embodiments, other methods of storage may be used. Proceeding to state 1250, the process verifies whether the user still wants a license. If the user's response is no, then the process proceeds to the end state 1255. Otherwise if the user indicates that the user still wants a license, then  
20       the process proceeds to state 1260, wherein the user computer 115 re-requests a license and proceeds to state 1270 wherein the user computer 115 finds the license and proceeds to state 1280. In state 1280, the user computer 115 finds the content and then proceeds to the end state 1290.

#### F.       User Manager Invokes Application

Once the user manager 320 has the license and the content, the user manager 320, in state 870, invokes the  
25       appropriate application with the requested content and proceeds to the end state 880. The user manager 320 allows the appropriate access to content according to the terms of the rules. For example, if the requested content was a song in MP3 format and the rules allowed the user computer 115 to play the first 20 seconds of the song, the user manager 320 would invoke an MP3 player on the user computer 115 with the requested song and allow the song to play for 20 seconds. The rules may allow for varying types of access to content such as viewing a portion of the  
30       content, viewing the entire content, copying a portion of the content, copying the entire content, printing a portion of the content, printing one full copy of the entire content, printing two full copies of the entire content, and so forth.

#### G.       Other Embodiments

While one embodiment allows the content to be requested before a license is created, it is recognized that in other embodiments the license could be created and/or requested before the content is created and/or requested. For

example, a user could purchase a license for a subscription to all new articles published in a specific publication. The user would receive a license for content that may not have been created. Moreover, it is recognized that a single piece of content could be related to one or more licenses and a single license could correspond to one or more pieces of content.

5           In one embodiment, the content manager 124 may also send the user a content location cell. The content location cell provides the user with information on where the content is located. The user may still need to receive a content cell 210 when the user is ready to access the content. For example, a user may want to view a video file of a two hour movie. Rather than send the entire piece of content to the user, a license management system module, such as the content manager 124, may send the user a content location cell giving the user access to the location of the  
10           requested content.

          The content location cell may be similar to a content cell, but contains an encrypted content location, such as a URL, rather than the content itself. The content location cell may be built by the content manager 124 or another module. Once the user receives the content location cell, the user may request a license to gain access to the content location. After the user receives a license to access the content location, the user may gain access to the content  
15           location. With this access the user may, for example, be able to see a snapshot of a piece of content, access advertising material, and so forth. The user may then request access to the content itself and, if required, request the appropriate license.

20

## V. Building The Cells

Figures 13 and 14 illustrate one embodiment for building content cells 210 and license cells 220. As previously discussed, it is recognized that in other embodiments, other modules could build the cells and/or the cells could be built using other techniques.

### 25           A. Building The Content Cell

A flowchart corresponding to one embodiment of building of the content cell 210 is illustrated in Figure 13. In the start state 920, the process proceeds to state 1310 wherein the content manager 124 extracts the rules 612. As previously discussed, the rules 612 tell what types of licenses and/or users can access various types of data and what is required for accessing the content. The rules 612 may be stored in a rules database or they may be listed as a  
30           text document and/or program wherein the content manager 124 can decipher and determine which rules 612 apply to the selected piece of content.

Proceeding to state 1320, the content manager 124 obtains content from the content database 438. In one embodiment, all of the content is stored in one database and the content manager 124 can search the database and retrieve the requested content.

Proceeding to state 1330, the content manager 124 then encrypts the content. The encryption technique may use a variety of techniques well known in the art. However, as previously discussed, in one embodiment the content manager 124 encrypts the content using a symmetric key and then encrypts the symmetric key with the offer manager's public key. When the encrypted content 614 and encrypted symmetric key are sent to the user computer  
5 115, the user computer 115 cannot decrypt the content and cannot decrypt the symmetric key until it has the appropriate license.

Proceeding to state 1340, the content manager 124 creates the cell header 616. The cell header 616 may include information such as a unique identifier for the piece of content, the store from which the content was derived, descriptive text, information about expiration, as well as other information.

10 Proceeding to state 1350, the content manager 124 then adds the extracted rules 612, the encrypted content 614, cell header 616, as well as its own public certificate 618 to the content cell 210. In state 1360, the content manager 124 "signs" the content cell 210. As previously discussed, the module that signs the content cell 210 must also include that module's public certificate in the cell. Thus, in state 1350, the content manager 124 includes its public certificate so that the recipient can verify that the content manager 124 was the one who sent and  
15 created the content cell 210. After the content manager 124 signs the content cell 210, the process proceeds to the end state 1370.

#### B. Building The License Cell

A flowchart corresponding to one embodiment of building of the license cell 220 is illustrated in Figure 14. In the start state 1150, the process proceeds to state 1410 wherein the license manager 530 extracts the rules 712. As  
20 previously discussed, the rules 712 tell what types of licenses and/or users can access various types of data and what is required for accessing the content. The rules 712 may be stored in a rules database or they may be listed as a text document and/or program wherein the license manager 530 can decipher and determine which rules 712 apply.

Proceeding to state 1420, the license manager 530 creates a license. Proceeding to state 1430, the license manager 530 then encrypts the license. The encryption technique may use a variety of techniques well known in the  
25 art. However, as previously discussed, in one embodiment the license manager 530 encrypts the license using a symmetric key and then encrypts the symmetric key with the user's public key. When the encrypted license 714 and encrypted symmetric key are sent to the user computer 115, the user computer 115 can decrypt the license.

Proceeding to state 1440, the license manager 530 creates the cell header 716. The cell header 716 may include information such as a unique identifier for the license cell, unique identifiers for related pieces of content, the  
30 store from which the content was derived, descriptive text, information about expiration, as well as other information.

Proceeding to state 1450, the license manager 530 then adds the extracted rules 712, the encrypted license 714, cell header 716, as well as its own public certificate 718 to the license cell 220. In state 1460, the license manager 530 "signs" the license cell 220. As previously discussed, the module that signs the license cell 220 must also include that module's public certificate in the cell. Thus, in state 1460, the license manager 530 includes its

public certificate so that the recipient can verify that the license manager 530 was the one who sent and created the license cell 220. After the license manager 530 signs the license cell 220, the process proceeds to the end state 1470.

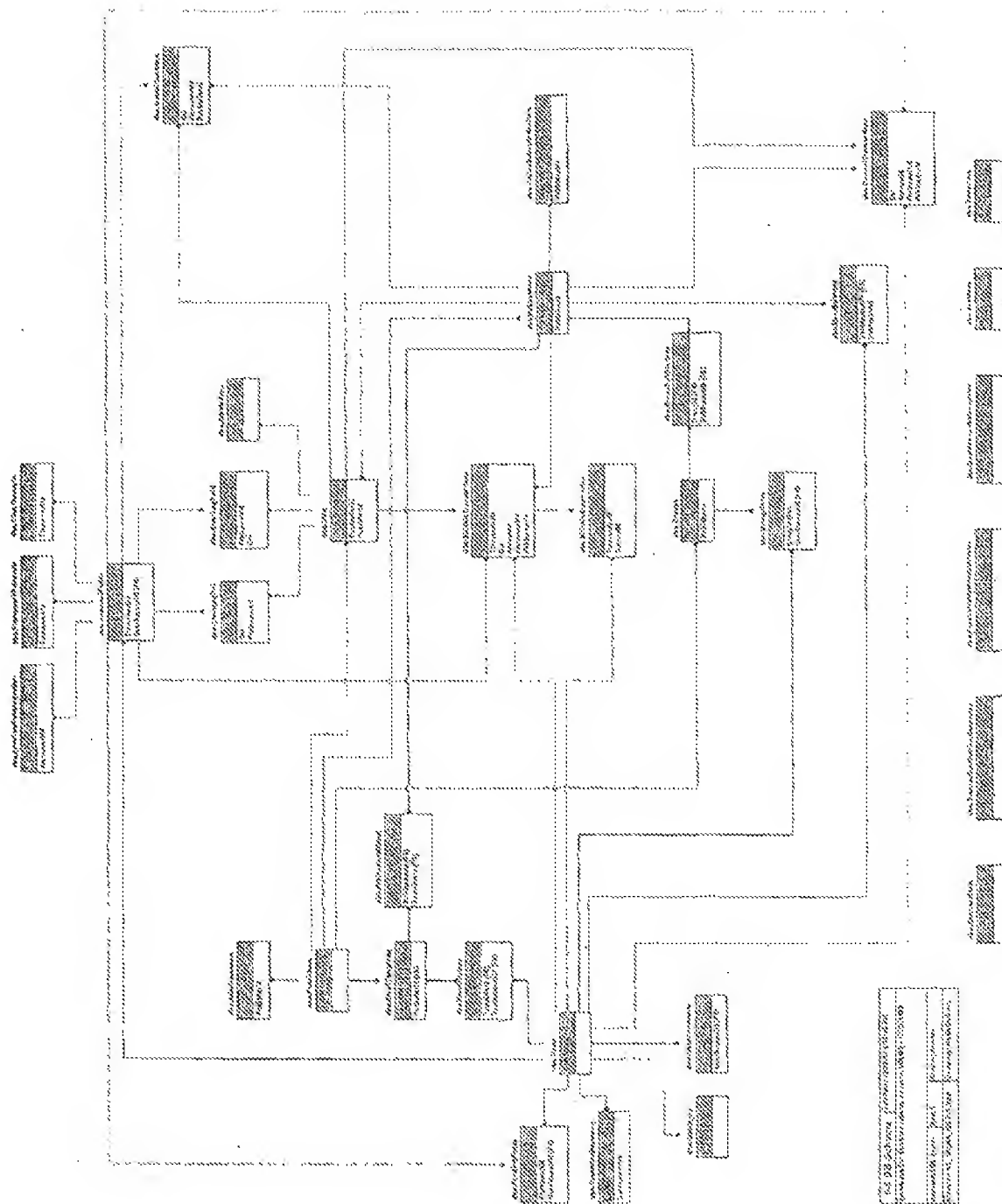
In one embodiment, the license manager 530 may store a copy of the license cell information in a license  
5 database 547 for later tracking the issued license.

#### VI. Conclusion

While certain preferred embodiments of the invention have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the present invention. Accordingly, the  
10 breadth and scope of the present invention should be defined only in accordance with the following claims and their equivalents.



# Appendix A



## ActivatedFwids Table

Registers every successful MediaCenter installation by a user. An activated Framework assigned to every user. Currently, there is a one-to-one relationship between a user and an activated Framework. Plans exist to change this relationship, so that one user can have many activated Frameworks.

Before a Framework is activated, it needs to first be pending. At activation time, a record for a Framework in the PendingFwids table is copied into the Activated table. At this point, the backend can tell who the user is and which Framework is being used.

ActivatedFwids									
Timestamp	datetime	8	0	0					(not delet.)
FrameworkId	varchar	255	0	0					
CellId	varchar	255	0	0					
Email	varchar	255	0	0					
RequestId	varchar	255	0	0					( )
UserHost	varchar	255	0	0					
RemoteHost	varchar	255	0	0					
Authentication	bit	1	0	0					(0)
UserAccountId	int	4	10	0					
ValidThrough	datetime	8	0	0					

### Field Descriptions

#### TimeStamp

Date and time when a Framework was activated.

#### FrameworkId

Framework Unique Identifier (e.g., fwid0001.0000001101). At this time, one Framework corresponds to one User Account Id; plans exist to allow a User Account Id to have multiple active Framework Ids.

#### CellId

Character string that uniquely identifies a Framework Id Cell. Every Cell, including Framework Id Cells, Contract Cells, and Content Cells, has a unique Id.

#### Email

Email account provided by a User at the time MediaCenter was installed.

#### RequestId

Verifies which Framework it's coming from; provides extra security (i.e., if backend

ActivatedFwids Table

Page 2 of 2

respond, it will try to activate and sends request Id; used to prevent hackers).

**UserHost**

No longer used.

**RemoteHost**

No longer used.

**Authentication**

Boolean value. False indicates that the Email address has not been authenticated. indicates the user has received the backend's confirmation Email and has clicked o link to complete the registration. This process authenticates, or verifies, that the E address is legitimate, because the backend knows that the user actually received t Email.

**UserAccountId**

Same as User.

**ValidThrough**

When the Framework Id expires (e.g., six months, one year, etc.).

## Administrators Table

Contains the list of special users that are authorized to perform administrative task. Administrators in this list have no relationship to Users. The Administrator logs in a privileged MediaDNA employees and publishers to manage their KnowledgeStor da

Administrators Table									
Name	varchar	50	0	0					
Password	varchar	40	0	0					
Email	varchar	100	0	0					
Phone	varchar	20	0	0					
PublisherId	int	4	10	0					
Type	varchar	1	0	0		(%)			
ManageCorpAccounts	bit	1	0	0		(1)			
ManagePhoneEmail	bit	1	0	0		(0)			

### Field Descriptions

#### Name

Name or login account for an Administrator.

#### Password

Password to gain access to the Administrator account.

#### Email

Email address of the Administrator. It allows electronic communication with the Administrator (i.e., send reports, statistics, etc.).

#### Phone

Telephone number where the Administrator can be contacted.

#### PublisherId

Publications this Administrator is entitled to manage.

#### Type

Defines this Administrator type. Currently only two types exists: Regular and Super. A Regular Administrator can only work with the designated publisher and its respective publications. A Super Administrator does not have this limitation. In other words, being a Super Administrator invalidates the PublisherId column. Valid values for this column are **R** and **S**.

#### ManageCorpAccounts

Indicates if this Administrator is allowed to manage corporate accounts.

Administrators Table

Page 2 of 2

**ManagePromoEmail**

Indicates if this Administrator is allowed to manage promotional Email.

## ArticleSync Table

Represents a queue for DOIs (articles) to be updated on the Dynamics database. Its purpose is to synchronize the DOI list in the Dynamics database with the DOI list available on the KnowledgeStor database.

ArticleSync									
Doi	varchar	255	0	0					
Operation	char	2	0	0					
Status	char	2	0	0		(P)			
ErrorReason	varchar	50	0	0		(E)			

### Field Descriptions

#### Doi

Doi of an article that needs to be synchronized.

#### Operation

Describes what to do to synchronize an article between databases. Currently, two operations are allowed: create and update. Valid values are **C** and **U**.

#### Status

Describes the status of the operation. Three values are defined: pending, in process, and error. Valid values are **P**, **I**, and **E**.

#### ErrorReason

If an error occurs during synchronization, the Status is set to **E** and the error message is stored in this field.

## BillableSummaries Table

Lists a condensed summary of all transactions made by a particular Framework (U) the billing information for that account. One entry reflects the total of many individ transactions in the BillableTransactions table.

BillableSummaries									
Field	Type	Size	Null	Default	Index	PK	FK	Ref	Comments
TransactionId	int	4	10	0					
TimeStamp	datetime	8	0	0					(nullable)
AccountId	int	4	10	0					
FirstName	varchar	60	0	0					
LastName	varchar	60	0	0					
Company	varchar	100	0	0					
Email	varchar	100	0	0					
AccountStatus	smalldint	2	5	0					(0)
ServiceLevel	smalldint	2	5	0					(0)
NameForBilling	varchar	100	0	0					
BillingAddress1	varchar	50	0	0					
BillingAddress2	varchar	50	0	0					
City	varchar	50	0	0					
State	varchar	40	0	0					
PostalCode	varchar	20	0	0					
Country	varchar	40	0	0					
CardType	varchar	10	0	0					
EncryptedCard	varchar	255	0	0					
EncryptionType	smalldint	2	5	0					(1)
CardExpirationY	varchar	2	0	0					
CardExpirationM	varchar	4	0	0					
Description	varchar	255	0	0					
CountOfTransact	int	4	10	0					
Amount	money	8	19	4					
Comment	varchar	255	0	0					

### Field Descriptions

#### TransactionId

Uniquely identifies a batch of transactions (one or many) for a User.

#### TimeStamp

Date and time when the transaction occurred.

#### AccountId

User performing the transaction.

#### FirstName

First name of the User.



BillableSummaries 1800

Page 2 of 3

**LastName**

Last name of the User.

**Company**

Company of the User. This field might be null.

**Email**

Email address of the User.

**AccountStatus**

Defines the status of the User account. A User account can be inactive (0) or can belong to a guest (1), a member (2) or a group member (3). Also, it can be disabled (4).

**ServiceLevel**

Intended for future use, to provide for any number of different layers of service provided to users (e.g., "gold," "silver," "elite," etc.). Currently there are only two levels: **Guest** and **Member**.

**NameForBilling**

Name used in billing operations for this User.

**BillingAddress1**

Billing address field #1.

**BillingAddress2**

Billing address field #2.

**City**

City for billing.

**State**

State for billing.

**PostalCode**

Postal code for billing.

**Country**

Country for billing.

**CardType**

Credit card type used for billing (e.g., Visa, MasterCard, etc.).

**EncryptedCardNumber**

Encrypted credit card number. It is encrypted for security reasons.

**EncryptionType**

Type of encryption used to encrypt the credit card number. Available

file:///F:/Technical Publications/TW Documents/Intranet Documents/.../BillableSummaries.htm 12/22/99

BillableSummaries Table

Page 3 of 3

types are Basic (1), RSA (2), and SimpleHash (3).

**CardExpirationMonth**

Credit card expiration month.

**CardExpirationYear**

Credit card expiration year.

**CountOfTransactions**

Number of transactions grouped in this summary.

**Amount**

Amount in dollars for the grouped transactions into this summary.

**Comment**

General comments field. Almost never used.

BillableTransactions Table

Page 1 of 2

## BillableTransactions Table

Lists individual billable transactions that occurred on KnowledgeStor. One transact occurs each time a contract is issued.

BillableTransactions									
Field	Type	Length	Nullable	Default	Index	PK	FK	Ref	Comments
TransactionId	int	4	NO	0					
TimeStamp	datetime	8	NO	0					(getdate())
TransTime	datetime	8	NO	0					
Description	varchar	255	NO	0					
Doc	varchar	255	NO	0					
AccountId	int	4	NO	0					
FrameworkId	varchar	255	NO	0					
PublicationId	int	4	NO	0					
TransType	int	4	NO	0					
Amount	money	8	NO	0					
Comment	varchar	255	NO	0					

### Field Descriptions

#### TransactionId

Identifies the transaction as part of a batch of transactions for one User.

#### TimeStamp

Date and time when the entry to this table was made. Generally, this is the time when the user encountered the "payment" page and authorized charges to a credit card.

#### TransTime

Date and time when the transaction took place; this may be hours, days, or weeks prior to the TimeStamp. Generally, it is the time when the user purchased each document.

#### Description

Describes what the transaction was for. It usually contains the title of the article purchased or informs about introductory credits.

#### Doc

Article involved in the transaction.

#### AccountId

User making the transaction.

#### FrameworkId

file://P:\Technical Publications\TW Documents\Intranet Documents\ABillableTransactions.htm 12/22/99

BillableTransactions Table

Page 2 of 2

FrameworkId identification number used by the User.

**PublicationId**

Publication the article belongs to.

**TransType**

Transaction type identified by a number. The valid types are Used (0), Introductory credit (1), Purchase (2), and Summary (3).

**Amount**

Amount in US dollars involved in this transaction.

**Comment**

General comments about the transaction. Usually this field is null.

ClickThroughs Table

Page 1 of 1

## ClickThroughs Table

Keeps track of each time a user clicks on KnowledgeStar to get to a published web

ClickThroughs									
Timestamp	datetime	8	0	0		(getdate())			
FrameworkId	varchar	255	0	0					
PublicationCode	varchar	255	0	0					

### Field Descriptions

#### TimeStamp

Date and time when the click through occurred.

#### FrameworkId

Framework that did the click through. This is later tracked to the User using this Framework.

#### PublicationCode

Publication in which the User completed the click through.

## ContractStats Table

When the backend is turned off, but before exiting, it logs the number of contracts and the average time it took to issue each one. These records are kept here. This not have a direct relationship with any other table in the backend Stor database. I for debugging and system monitoring.

ContractStats									
Timestamp	datetime	8	0	0				(getdate())	
NumberContracts	int	4	10	0					
AverageTime	int	4	10	0					
Comment	varchar	255	0	0				(null)	

### Field Descriptions

#### TimeStamp

Date and time when the backend finished execution.

#### NumberContracts

Number of contracts created (issued) by the backend from the time it was last initialized to the time when it was gracefully taken down.

#### AverageTime

Average time for creating a contract. This value is used to monitor the performance of the contract creation code.

#### Comment

General comments. Usually shows a message telling that the backend is finishing execution.

CreditCardNumbers Table

Page 1 of 1

## CreditCardNumbers Table

Holds an encrypted version of one credit card number for each user. The encrypted credit card number is also specified.

CreditCardNumbers									
	UserAccountId	int	4	10	0				
	EncryptionType	smallint	2	9	0		(1)		
	EncryptedCardNumber	varchar	255	0	0				

### Field Descriptions

#### UserAccountId

Identifies the account this card number is good for.

#### EncryptionType

Method used to encrypt credit card number. Available types are Basic (1), RSA (2), and SimpleHash (3).

#### EncryptedCardNumber

Encrypted credit card number.

CurrentTransactions Table

Page 1 of 2

## CurrentTransactions Table

Saves information related to transactions. An entry to the CurrentTransactions tab made for each contract. An entry can be a debit or credit and these entries may be purged. Each time a user buys an article, you get an IssuedContract entry and a CurrentTransaction entry.

CurrentTransactions Table									
	Field Name	Field Type	Length	Index	Null	Default	PK	FK	Ref
*	TimeStamp	datetime	8	0	0	((getdate()))			
	Description	varchar	255	0	0				
	Doi	varchar	255	0	0				
	AccountId	int	4	10	0				
	FrameworkId	varchar	255	0	0				
	PublicationId	int	4	10	0				
	TransType	int	4	10	0				
	Amount	money	8	19	0				
	Comment	varchar	255	0	0				

### Field Descriptions

#### TimeStamp

Date and time when the transaction occurred.

#### Description

Describes what the transaction was for. It usually contains the title of the article purchased or informs about introductory credits (i.e., if transaction type is purchase, the name of the article is entered).

#### Doi

Article involved in the transaction.

#### AccountId

Identifies user.

#### FrameworkId

Identifies which Framework Id is responsible for handling this transaction.

#### PublicationId

Identifies the publication of the involved article.

#### TransType

Transaction type identified by a number. The valid types are Unused (0), Introductory credit (1), Purchase (2), and Summary (3).



CurrentTransactions Table

Page 2 of 2

**Amount**

Amount in US dollars for this transaction.

**Comment**

General comments field.

file://F:\Technical Publications\TW Documents\Intranet Document...CurrentTransactions.htm 12/22/99

DOIs Table

Page 1 of 2

## DOIs Table

Accommodates article-related information. An article is also known as a DOI. Ever represents an available DOI in the KnowledgeStor. This table is an article inventor contains all articles a User would access, all Content Cells, and more. This table is the naming servlet.

DOI									
Doi	varchar	255	0	0					
TimeStamp	datetime	8	0	0				(getdate())	
Title	varchar	255	0	0				(')	
PublicationId	int	4	10	0					
PublisherId	int	4	10	0					
Url	varchar	255	10	0					
Price	money	8	10	4					
ValidFrom	datetime	8	0	0				(getdate())	
ValidThrough	datetime	8	0	0				(12/31/2199)	
ContentCell	bit	1	0	0				(1)	

### Field Descriptions

#### TimeStamp

Date and time a DOI was registered.

#### Title

Title of the article.

#### PublicationId

Publication Identification number this DOI belongs to.

#### PublisherId

Publisher Identification number this DOI belongs to.

#### Url

Address where article cell is located (e.g.,  
file:///d:/dataDirectory/cells/com.cadcamnet\_cad\_018\_01\_01.mdna).

#### Price

Amount in US dollars that this article is priced at.

#### ValidFrom

Indicates when this DOI can first be accessed. Limits access to articles.

#### ValidThrough

Indicates when this DOI can be accessed until. Limits access to articles.

file://PATechnical Publications\TW Documents\Intranet Documentation\Database\DOIs.htm 12/22/99

DOIs Table

Page 2 of 2

**ContentCell**

Indicates if the cell identified by this DOI is a Content Cell. Values are True(1) or False(0).

file://F:\Technical Publications\TW Documents\Intranet\Documentation\Database...DOIs.htm 12/22/99

EmailHistory Table

Page 1 of 1

## EmailHistory Table

Keeps track of Emails sent to particular Users. The date when the Email was sent a template used is also recorded.

EmailHistory									
TemplateId	int	4	10	0					
DateSent	datetime	8	0	0					(getdate())
ToHeader	nvarchar	100	0	0					
UserAccountId	int	4	10	0					

### Field Descriptions

#### TemplateId

Email template used. Foreign key to EmailTemplates.TemplateId.

#### DateSent

Date when the Email was sent.

#### ToHeader

Email accounts to which the Email was sent. The User can modify this data in their customer profile, so we store the address that was actually used.

#### UserAccountId

User receiving the Email. Foreign key to Users.AccountId.

## EmailTemplates Table

This table houses the collection of templates the system can use to generate Email messages. To generate an Email message, the system takes the **FromHeader**, **SubjectHeader**, and **Body** from the database entry and uses them to build the message. Templates are either owned by a specific publisher or by the system.

EmailTemplates									
Field Name	Field Type	Field Length	Field Null	Field Default	Field Index	Field Unique	Field Primary	Field Foreign	Field Comment
TemplateId	int	4	NO	0					
PublisherId	int	4	NO	0					
Name	varchar	255	NO						
CreationDate	datetime	8	NO						(getdate())
SubjectHeader	varchar	100	NO						
FromHeader	varchar	100	NO						
Body	varchar	4000	NO						

### Field Descriptions

#### TemplateId

Uniquely identifies template by this number.

#### PublisherId

Publisher that owns this template. PublisherId of 0 (zero) indicates a system template.

#### Name

Name of this template; this name is unique among all templates owned by this publisher.

#### CreationDate

Date when the template was created.

#### SubjectHeader

Subject line when sending an Email with this template.

#### FromHeader

Defines the FromHeader for Emails sent using this template.

#### Body

Contents of the Email using this template.

ErrorLogDoi Table

Page 1 of 1

## ErrorLogDoi Table

Records Doi-related errors. The date when it occurred, the type of error, the Fram (User) to whom it was issued, and the attempted Doi are part of this log.

ErrorLogDoi									
Timestamp	datetime	8	0	10					(getdate())
ErrorNumber	smallint	2	5	0					
Doi	varchar	255	0	10					()
FrameworkId	varchar	255	0	10					()
RemoteAddr	varchar	40	10	0					()
RemoteHost	varchar	90	0	10					()

### Field Descriptions

#### TimeStamp

Date and time when the error was logged.

#### ErrorNumber

Error number describing the failure. Common Doi errors are a missing Doi (101) or unresolved Doi (102).

#### Doi

Intended access to this article caused an error.

#### FrameworkId

Framework requesting the failing article.

#### RemoteAddr

Remote IP address to which the Doi error was issued.

#### RemoteHost

Remote host name to which the Doi error was issued.

ErrorLogFwid Table

Page 1 of 1

## ErrorLogFwid Table

Records FrameworkId-related errors. The date when it occurred, the type of error, Framework (User) to whom it was issued, and the attempted Doi are part of this log.

ErrorLogFwid									
TransactionId	int	4	10	0					
TimeStamp	datetime	8	0	0			(datetime)		
ErrorNumber	smallint	2	5	0					
FrameworkId	varchar	255	0	0					
Doi	varchar	255	0	0			( )		
RemoteAddr	varchar	40	0	0			( )		
RemoteHost	varchar	40	0	0			( )		
ExtraText	varchar	255	0	0			( )		

### Field Descriptions

#### TransactionId

Unique transaction number identifying the error.

#### TimeStamp

Date and time when the error was logged.

#### ErrorNumber

Number describing the error. Possible errors include Fwid missing (2001), in bad format (2002), unregistered (2003), no account found (2004), not activated (2005), and expired (2006).

#### FrameworkId

Framework involved in the error.

#### Doi

Doi involved in the error.

#### RemoteAddr

User IP address involved in the error.

#### RemoteHost

User host name involved in the error.

#### ExtraText

General comments.

file://F:\Technical Publications\TW Documents\Intranet Documentation\.. \ErrorLogFwid.htm 12/22/99

ErrorLogFwidActivation Table

Page 1 of 1

## ErrorLogFwidActivation Table

Records FrameworkId activation-related errors.

ErrorLogFwidActivation									
Timestamp	datetime	8	0	0				(getdate())	
FrameworkId	varchar	255	0	0				()	
CellId	varchar	255	0	0				()	
Expires	varchar	255	0	0				()	
UserHost	varchar	255	0	0				()	
RequestId	varchar	255	0	0				()	
RemoteAddr	varchar	40	0	0				()	
RemoteHost	varchar	40	0	0				()	

### Field Descriptions

#### Timestamp

Date and time when the error was logged.

#### FrameworkId

Framework involved in the error.

#### CellId

Character string which uniquely identifies the Framework Id Cell experiencing the error. Every Cell, including Framework Id cells, Contract Cells, and Content Cells, has a unique Id.

#### Expires

Time when the Framework Id Cell is to expire.

#### UserHost

Column will no longer be used.

#### RequestId

Request number.

#### RemoteAddr

User IP address.

#### RemoteHost

Column will no longer be used.

file://F:\Technical Publications\TW Documents\Intranet Docs...ErrorLogFwidActivation.htm 12/22/99



FrameworkReinstalls Table

Page 1 of 1

## FrameworkReinstalls Table

Counts the number of Framework reinstalls attempted by a User, and the new Framework Identification number assigned after each successful reinstall.

FrameworkReinstalls									
Timestamp	datetime	8	0	0		(getdate())			
FrameworkId	varchar	255	0	0					
InstFwid	varchar	255	0	0					
ChainLength	int	4	16	0					
RemoteUser	varchar	255	0	0					
RemoteAddr	varchar	40	0	0					
RemoteHost	varchar	40	0	0					

### Field Descriptions

#### TimeStamp

Date when a Framework reinstall was logged.

#### FrameworkId

Framework previously installed.

#### InstFwid

Reinstall Framework.

#### ChainLength

Number of Framework reinstalls attempted by the User. When the count reaches a predefined limit, no more reinstalls are allowed for the User.

#### RemoteUser

Column will no longer be used.

#### RemoteAddr

IP address of the user reinstalling a Framework.

#### RemoteHost

Host name of the user reinstalling a Framework.

FwidCellidNumberGenerator Table

Page 1 of 1

## FwidCellidNumberGenerator Table

This is the mechanism by which unique cell numbers are created. Keeps track of a Cell Id number. This table does not have a direct relationship with any other table backend KnowledgeStor database.

FwidCellidNumberGenerator									
NextCellidNumber	Int	4	10	0					

### Field Description

#### NextCellidNumber

Keeps track of the next cell identification number.

GroupPublications Table

Page 1 of 1

## GroupPublications Table

Identifies which publications can be accessed by any of the User Group members.

GroupPublications									
PK	GroupId	Int	4	10	0				
	PublicationId	Int	4	10	0				

### Field Descriptions

#### GroupId

Group Identification number.

#### PublicationId

Publication Identification number.

Groups Table

Page 1 of 1

## Groups Table

Maintains group-related information. Groups are used to accomplish corporate acc users can be consolidated into a single account using corporate accounts.

Field	Type	Length	Nullable	Default	Comments
GroupId	int	4	NO	0	
Name	varchar	50	NO		
Password	varchar	40	NO		
Type	varchar	1	NO		(D) (S)
UserTTL	int	4	NO	0	(365)
PublicationId	int	4	NO	0	
Expires	datetime	8	NO		(dateadd(year,1,getdate()))

### Field Descriptions

#### GroupId

Group identification number. This number is unique.

#### Name

Name of the group.

#### Password

Access password for accounts under this group.

#### Type

Type of valid group accounts. Allowed group accounts are demo accounts: D, and standard accounts: S. Users in a demo account can be promoted to a standard account if respective group account fees are paid.

#### UserTTL

Number of days group accounts will be valid for this group.

#### PublicationId

Publication members of this group have access to.

#### Expires

Expiration date for this group account.

GroupUsers Table

Page 1 of 1

## GroupUsers Table

Relates Users to a Group. Identifies the members of a Group.

GroupUsers					
GroupId	int	4	10	0	
UserAccountId	int	4	10	0	
Expires	datetime	8	10	0	

### Field Descriptions

#### GroupId

Group identification number.

#### UserAccountId

User account identification number.

#### Expires

Date when this Group account expires for this User.

IssuedContracts Table

Page 1 of 2

## IssuedContracts Table

Records every contract issued to a particular user when getting an article. One entry made for every contract issued to a Framework (one contract for each article issue). These entries are kept permanently so a user won't be charged again for download same article.

IssuedContracts									
TransactionId	int	4	10	0					
TimeStamp	datetime	8	0	0			(datetime)		
Doi	varchar	255	0	0					
ContractUrl	varchar	255	0	0					
CellId	varchar	255	0	0					
FrameworkId	varchar	255	0	0					
PublicationId	int	4	10	0					
Amount	money	8	19	4					

### Field Descriptions

#### TransactionId

Unique number and identity for this transaction for auditing.

#### TimeStamp

Date and time when this transaction was registered.

#### Doi

Tells which article is for this contract.

#### ContractUrl

Identifies where the Contract Cell is located. (The Contract Cell is discarded after a while to keep the disk clean.)

#### CellId

Character string which uniquely identifies the Contract Cell. Every Cell, including Framework Id Cells, Contract Cells, and Content Cells, has a unique Id.

#### FrameworkId

Framework Id has permission to access the Cell identified by the Doi.

#### PublicationId

Tells which publication the Doi in this contract belongs to.

#### Amount

file://PATechnicalPublications/TW Documents/Internet Documentation/IssuedContracts.htm 12/22/99

IssuedContracts Table

Page 2 of 2

Amount User was charged; used for debugging and tracking.

OnlineSubscriptionDefs Table

Page 1 of 1

## OnlineSubscriptionDefs Table

Contains available definitions for online subscriptions managed by KnowledgeStor.

OnlineSubscriptionDefs					
Description	varchar	255	0	0	
PublicationId	int	4	10	0	
Price	money	8	19	4	

### Field Descriptions

#### Description

Describes the characteristics of an online subscription.

#### PublicationId

Publication that supports online subscription.

#### Price

Price for the online subscription for this publication under this description.



PendingFwids Table:

Page 1 of 1

## PendingFwids Table

Lists all Frameworks pending. A pending Framework usually belongs to a guest. Use a PendingFwid is activated, its entry is "moved" from this table to the ActivatedFwi

PendingFwids									
Field	Type	Size	Null	Default	Index	Foreign Key	Relationship	Comments	
CellId	varchar	255	0	0					
FrameworkId	varchar	255	0	0					
Email	varchar	255	0	0					
ValidThrough	datetime	8	0	0					
UserAccountId	int	4	10	0					

### Field Descriptions

#### CellId

Character string which uniquely identifies the Framework Id Cell. Every Cell, including Framework Id Cells, Contract Cells, and Content Cells, has a unique Id.

#### FrameworkId

Framework pending.

#### Email

Email of the guest User with a pending Framework.

#### ValidThrough

Date this pending Framework will be valid until.

#### UserAccountId

User account holding this pending Framework.

PromotionalEmail Table

Page 1 of 1

## PromotionalEmail Table

Registers properties and characteristics of each promotional Email available on the backend.

PromotionalEmail									
PublicationId	int	4	10	10					
TemplateId	int	4	10	10					
Enabled	bit	1	10	10					
EnableDate	datetime	8	18	18					

### Field Descriptions

#### PublicationId

Publication involved in this promotional Email.

#### TemplateId

Template used for this promotional Email.

#### Enabled

Tells if this promotional Email is enabled.

#### EnableDate

Date when this promotional Email was enabled.

## Publications Table

Has information of every publication on the KnowledgeStar. One publication may contain many articles or DOIs.

Details (HTML)									
Field	Type	Length	Nullable	Default	Check	Comments	Index	PK	FK
PublicationId	int	4	NO	0					
Status	varchar	255	NO	0		('active')			
PublicationCode	varchar	255	NO	0					
ResourceFile	varchar	255	NO	0					
PublisherId	int	4	NO	0					
Domain	varchar	255	NO	0					
Name	varchar	255	NO	0					
ContentCellExtension	varchar	10	NO	0		('media')			
LibHomePage	varchar	255	NO	0					
IntroCredit	money	8	NO	4					
DefaultPrice	money	8	NO	4					
FreeArticles	int	4	NO	0					
PermitPrinting	bit	1	NO	0		(1)			
PermitCopying	bit	1	NO	0		(0)			

### Field Descriptions

#### PublicationId

Unique number for each publication. Identifies a publication internally.

#### Status

Status of a publication can be "active" or "inactive."

#### PublicationCode

Short code to identify a publication; usually it is a three-letter code (e.g., csn, olr, or cad).

#### ResourceFile

Points to a file where this publication's properties are found (e.g., c:\back\properties\ATContentCell.properties).

#### PublisherId

Publisher to which this publication belongs to.

#### Domain

Publisher domain for this publication (e.g., com.g2news, com.cadcamnet, etc.).

#### Name

Publications Table

Page 2 of 2

Name of the publication.

**ContentCellExtension**

Cell format for the content of this publication (e.g., MDNA, XDNA).

**UrlHomePage**

Web address for this publication (e.g.,  
<http://www.g2news.com/csnindex.html>).

**IntroCredit**

Introductory credit given to users accessing articles from this publication.  
The amount is in US dollars.

**DefaultPrice**

Default price for articles of this publication. The amount is in US dollars.

**FreeArticles**

Number of free articles given to users.

**PermitPrinting**

Indicates if printing is permitted for articles of this publication. Valid values are True (1) or False (0).

**PermitCopying**

Indicates if copying is permitted for articles of this publication. Valid values are True (1) or False (0).

Publishers Table

Page 1 of 1

## Publishers Table

Contains information from every publisher that has content in KnowledgeStor. One publisher may have many publications.

Publishers									
Field Name	Field Type	Length	Scale	Null	Default	Check	Comments	Index	PK
PublisherId	int	4	10	0					
Status	varchar	255	0	0			(active)		
PublisherName	varchar	255	0	0					
DefaultPrice	money	8	19	0					

### Field Descriptions

#### PublisherId

Publisher's unique identification number.

#### Status

Status of a publication can be "active" or "inactive."

#### PublisherName

Publisher name.

#### DefaultPrice

Default price for articles of this publisher. The amount is in US dollars.

TransactionNumberGenerator Table

Page 1 of 1

## TransactionNumberGenerator Table

This is the mechanism by which unique transaction id numbers are created. Keeps unique transaction id number. This table does not have a direct relationship with a table in the KnowledgeStor database.

TransactionNumberGenerator									
1	NextTransactionNumber	int	4	10	10				

### Field Description

#### NextTransactionNumber

Holds the next transaction number.

UserBalances Table

Page 1 of 1

## UserBalances Table

Keeps a record of a User's credit and charges for a specific publication. Entries exist for each publication a User has accessed.

UserBalances									
UserAccountId	int	4	10	0					
PublicationId	int	4	10	0					
TimeCreditCreated	datetime	8	0	0			:(getdate())		
AvailableCredit	money	8	19	4			(0.0)		
CreditApplied	money	8	19	4			(0.0)		
CurrentCharges	money	8	19	4			(0.0)		
HaveShownPricingTerms	bit	1	0	0			(0)		

### Field Descriptions

#### UserAccountId

User accessing an article.

#### PublicationId

Publication that the article being accessed belongs to.

#### TimeCreditCreated

Date and time when credit was created for this User on articles from this publication.

#### AvailableCredit

Amount of available credit in US dollars .

#### CreditApplied

Amount of credit applied in US dollars .

#### CurrentCharges

Amount of current charges in US dollars for this User on articles from this publication.

#### HaveShownPricingTerms

Indicates if a User has seen the terms and conditions for articles of this publication. Valid values are True (1) or False (0).

UserComments Table

Page 1 of 1

## UserComments Table

Holds comments made for each particular user. This table is used to aid in custom support.

UserComments									
	UserAccountId	int	4	10	0				
	TimeStamp	datetime	8	0	0			(getdate())	
	Comment	varchar	255	0	0				

### Field Descriptions

#### UserAccountId

User for which comments were registered.

#### TimeStamp

Date and time when the comments were registered.

#### Comment

General comments for this User.



Users Table

Page 1 of 3

## Users Table

Maintains information for a User account in KnowledgeStor. This information includes address, account status, credits, and charges for every User. There is one entry in for each User. Once a User does something that requests a Framework Id number is created and a Framework Id number is sent.

Users Table									
Field Name	Field Type	Field Length	Field Null	Field Default	Field Comments	Field Index	Field Primary	Field Foreign	Field Unique
UserAccountID	int	4	NO	0					
FirstName	varchar	60	NO						
LastName	varchar	60	NO						
Company	varchar	100	NO						
Email	varchar	100	NO						
AccountCreationDate	datetime	8	NO		{getdate()}				
AccountStatus	smallint	2	NO	0	(0)				
CurrentCharges	money	8	NO		(0.0)				
CurrentCredits	money	8	NO		(0.0)				
OutstandingLink	money	8	NO		(0.0)				
ServiceLevel	smallint	2	NO	0	(0)				
ProvisionalEmail	int	1	NO	0	(0)				
RemoteUser	varchar	255	NO						
RemoteAddr	varchar	40	NO						
RemoteHost	varchar	40	NO						
RemotePort	varchar	100	NO						
BillingAddress1	varchar	60	NO						
BillingAddress2	varchar	60	NO						
City	varchar	60	NO						
State	varchar	40	NO						
PostalCode	varchar	20	NO						
Country	varchar	40	NO						
Password	varchar	40	NO						
PasswordEncryption	smallint	2	NO	0	(0)				
CardType	varchar	10	NO						
CardAccountNumber	varchar	20	NO						
CardExpirationMonth	varchar	2	NO						
CardExpirationYear	varchar	4	NO						

### Field Descriptions

#### UserAccountID

Identifies each User.

#### FirstName

First name of the User.

#### LastName

Last name of the User.

## Users Table

Page 2 of 3

**Company**

Company of the User. This might be null.

**Email**

Email address of the User.

**AccountCreationDate**

Date when this User account was created.

**AccountStatus**

Status of this User account. A User account can be inactive (0) or can belong to a guest (1), a member(2), or a group member (3). It can also be disabled (4). A "guest" is one who has not registered fully with KnowledgeStor. A "member" has provided complete information including billing information.

**CurrentCharges**

Current charges in US dollars made to this User account.

**CurrentCredits**

Available credit in US dollars for this User.

**OutstandingLimit**

The limit that must be reached for a User to be asked to pay; can be set per User.

**ServiceLevel**

Intended for future use, to provide for any number of different levels of service offered to users (e.g., "gold," "silver," "elite," etc.). Currently there are two levels -- "Guest" and "Member."

**PromotionalEmail**

Indicates if this User wants to receive promotional Emails. Valid values are True (1) and False (0).

**RemoteUser**

Column will no longer be used.

**RemoteAddr**

User's IP address.

**RemoteHost**

Column will no longer be used.

**NameForBilling**

Name used for billing.

file://F:\Technical Publications\TW Documents\Intersnet Documentation\Database...Users.htm 12/22/99

Users Table

Page 3 of 3

**BillingAddress1**

User billing address field #1.

**BillingAddress2**

User billing address field #2.

**City**

City of the User.

**State**

State of the User.

**PostalCode**

Postal code of the User.

**Country**

Country of the User.

**Password**

User password.

**PasswordEncryptionType**

Type of encryption used to encrypt the User password. Available types are Basic (1), RSA (2), and SimpleHash (3).

**CardType**

Type of credit card (e.g., Visa, MasterCard, etc.).

**CardAccountNumber**

Credit card number that is safe to display to the user, such as "xxxxxxxxxxxx1234."

**CardExpirationMonth**

Credit card expiration month.

**CardExpirationYear**

Credit card expiration year.

UserSync Table

Page 1 of 1

## UserSync Table

Represents a queue for Users to be updated on the Dynamics database. Its main p is to synchronize the User list in the Dynamics database with the User list available KnowledgeStor database.

UserSync									
UserAccountId	int	4	10	0					
Operation	varchar	12	0	0					
Status	varchar	2	0	0		(P)			
ErrorReason	varchar	255	0	0		(E)			

### Field Descriptions

#### UserAccountId

User account to synchronize in both databases.

#### Operation

Describes what to do when synchronizing an article between databases. Currently two operations are allowed: **Create** and **Update**. Valid values are **C** and **U**.

#### Status

Describes the status of the operation. Three values are defined: pending, in process, and error. Valid values are **P**, **I**, and **E**.

#### ErrorReason

If an error is raised and the status is marked with an **E**, an error message will appear here.

WHAT IS CLAIMED IS:

1. A method of granting access to a data object comprising:  
creating a data object cell;  
sending the data object cell to a computing device over a network;  
5 creating a license object cell related to the data object cell;  
sending the license object cell to the computing device over the network; and  
granting access to the data object cell based upon information in the related license object cell.
2. The method of Claim 1, additionally comprising receiving a user request for a data object cell.
3. The method of Claim 1, additionally comprising receiving a user request for a license object.
- 10 4. The method of Claim 1, wherein granting access to the data object cell includes verifying that a license object cell related to the data object cell is located on the computing device.
5. The method of Claim 1, wherein the data object cell includes at least a data control element and a data object.
6. The method of Claim 5, additionally comprising encrypting at least the data object.
- 15 7. The method of Claim 1, wherein at least a portion of the license object cell is encrypted.
8. The method of Claim 1, additionally comprising:  
retrieving a data location cell related to the data object cell;  
sending the data location cell to a computing device over a network;  
granting access to the data location cell based upon information in a related license object cell.
- 20 9. A method of granting access to a data object comprising:  
retrieving a data object cell;  
sending the data object cell to a computing device over a network;  
retrieving a license object cell related to at least one data object cell;  
sending the license object cell to the computing device over the network; and  
25 granting access to the data object cell based upon information in a related license object cell.
10. The method of Claim 9, additionally comprising receiving a user request for a data object cell.
11. The method of Claim 9 additionally comprising receiving a user request for a license object.
12. The method of Claim 9, wherein granting access to the data object cell includes verifying that a license object cell related to the data object cell is located on the computing device.
- 30 13. The method of Claim 9, wherein the data object cell includes at least a data control element and a data object.
14. The method of Claim 13 additionally comprising encrypting at least the data object.
15. The method of Claim 9, wherein at least a portion of the license object cell is encrypted.
16. The method of Claim 9, additionally comprising creating a data object cell.

17. The method of Claim 9, additionally comprising creating a license object cell.

18. The method of Claim 9, wherein the data object cell and the license object cell are related by at least one unique identifier.

19. A method of managing data objects, the method comprising:

- 5            sending a request to a first computing device for a data object;  
             receiving a data object cell related to the requested data object from the first computing device;  
             sending a request to a second computing device for a license object related to the data object cell;  
             receiving a license object cell from the second computing device that is related to the requested  
license object; and  
10            granting access to the data object cell based on information in the related license object cell.

20. The method of Claim 19, wherein the first computing device and the second computing device are the same computing device.

21. A method of managing data objects comprising:

- receiving a request for a data object from a network node;  
15            identifying a data object cell that is associated with the requested data object;  
             sending the data object cell to the network node wherein access to the data object cell requires a  
related license object cell.

22. The method of Claim 21, additionally comprising:

- accessing a license object cell related to the data object cell;  
20            sending the license object cell to the network node wherein the license object cell grants access to  
at least the data object cell.

23. A method of managing data objects comprising:

- receiving a request for a license object from a network node;  
             accessing a license object cell related to the requested license object; and  
25            sending the license object cell to the network node wherein the license object cell grants access to  
at least one data object cell.

24. A network comprising:

- a user node coupled to the network, wherein the user node comprises a user manager module and  
provides requests for data objects and related license objects on the network;  
30            a content manager node coupled to the network, wherein the content server node receives requests  
for data objects on the network, retrieves requested data objects, and responds to the requests for data  
objects on the network; and

a license manager node coupled to the network, wherein the content server node receives requests for related license objects on the network, retrieves requested related license objects, and responds to the requests for related license objects on the network.

25. A network comprising:

- 5                   a content manager node responsive to request to provide a data object;  
                  a license manager node responsive to request to provide a license object related to at least one data object; and  
                  a user node providing access to the data object with the related license object.

26. The network of Claim 25, wherein the user node additionally providing:

- 10                   a user interface to a user for requesting one or more data objects;  
                  a request for selected data objects from the content manager node; and  
                  a determination of whether the user node has access to a data object and a license object.

27. A system for managing data objects comprising:

- a content module for receiving requests for data objects, retrieving the data objects, and  
15                   transmitting the requested data objects; and  
                  a license module for receiving requests for license objects related to the data objects, retrieving the license objects related to the data objects, and transmitting the requested license objects.

28. A system for managing data objects comprising:

- means for managing the data objects;  
20                   means for managing the license objects; and  
                  means for granting access to the data objects based at least upon a related license object.

29. A method of granting access to a data object comprising:

- data object cell retrieval means;  
                  data object cell transmission means;  
25                   license object cell retrieval means;  
                  license object cell transmission means; and  
                  access granting means.

30. A data object cell comprising:

- a data control element; and  
30                   a data object.

31. The data object cell of Claim 30 where in the data object is encrypted.

32. The data object cell of Claim 30 additionally comprising a cell header.

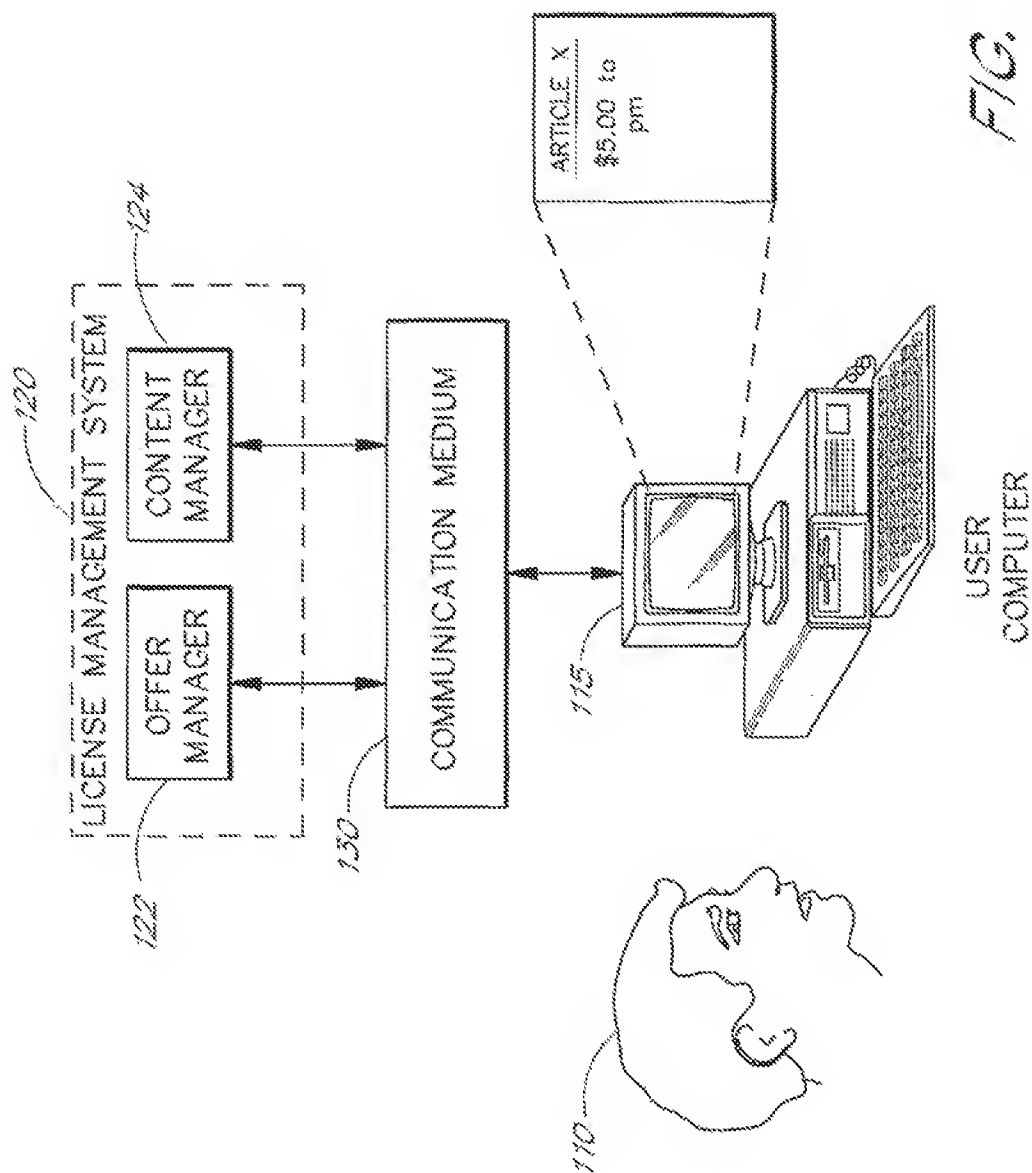
33. The data object cell of Claim 30 additionally comprising at least one public certificate.

34. The data object cell of Claim 30 additionally comprising at least one signature.

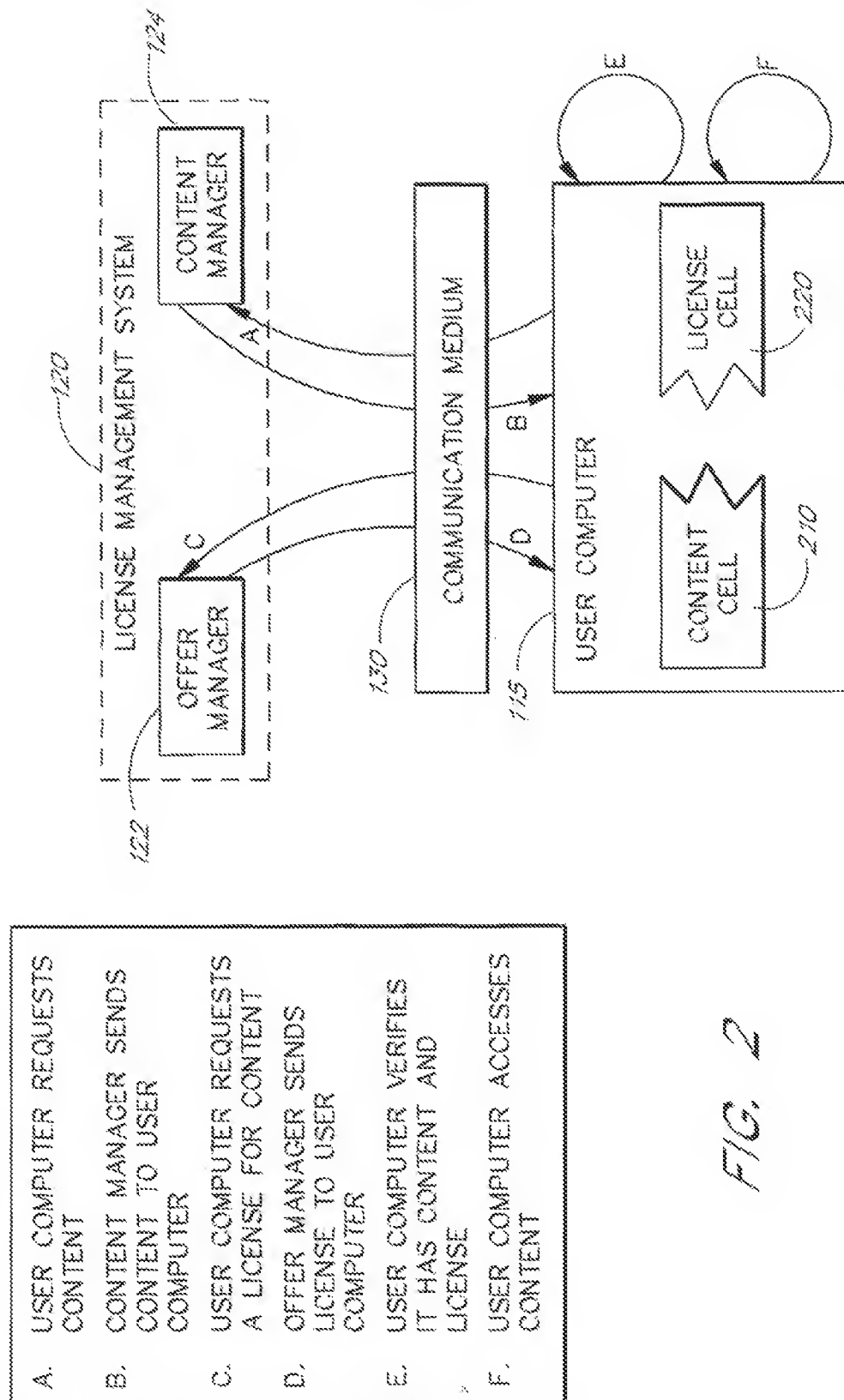
35. A license object cell comprising:  
a license control element; and  
a license object.
36. The license object cell of Claim 30 where in the license object is encrypted.
- 5 37. The license object cell of Claim 30 additionally comprising a cell header.
38. The license object cell of Claim 30 additionally comprising at least one public certificate.
39. The data object cell of Claim 30 additionally comprising at least one signature.
40. A database schema relating:  
user data;  
10 content data; and  
license data.
41. The database schema of Claim 40 wherein the user data comprises group data.
42. The database schema of Claim 40 wherein the content data comprises:  
publisher data;  
15 publication data; and  
article data.
43. The database schema of Claim 40 wherein the content data comprises subscription data.
44. The database schema of Claim 40 wherein the content data comprises billing data.
45. The database schema of Claim 40 wherein the license data comprises issued license data.



1/16



2/16



3/16

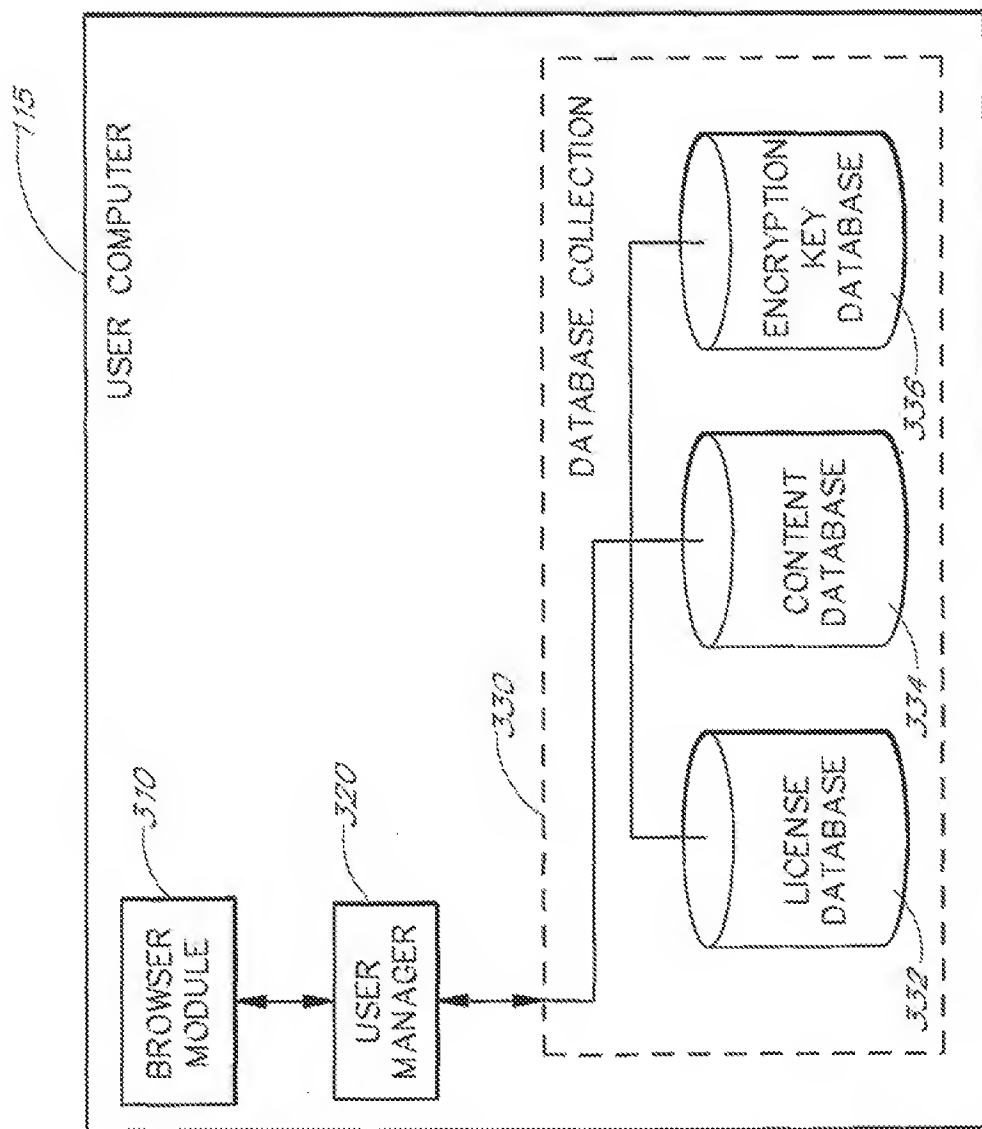
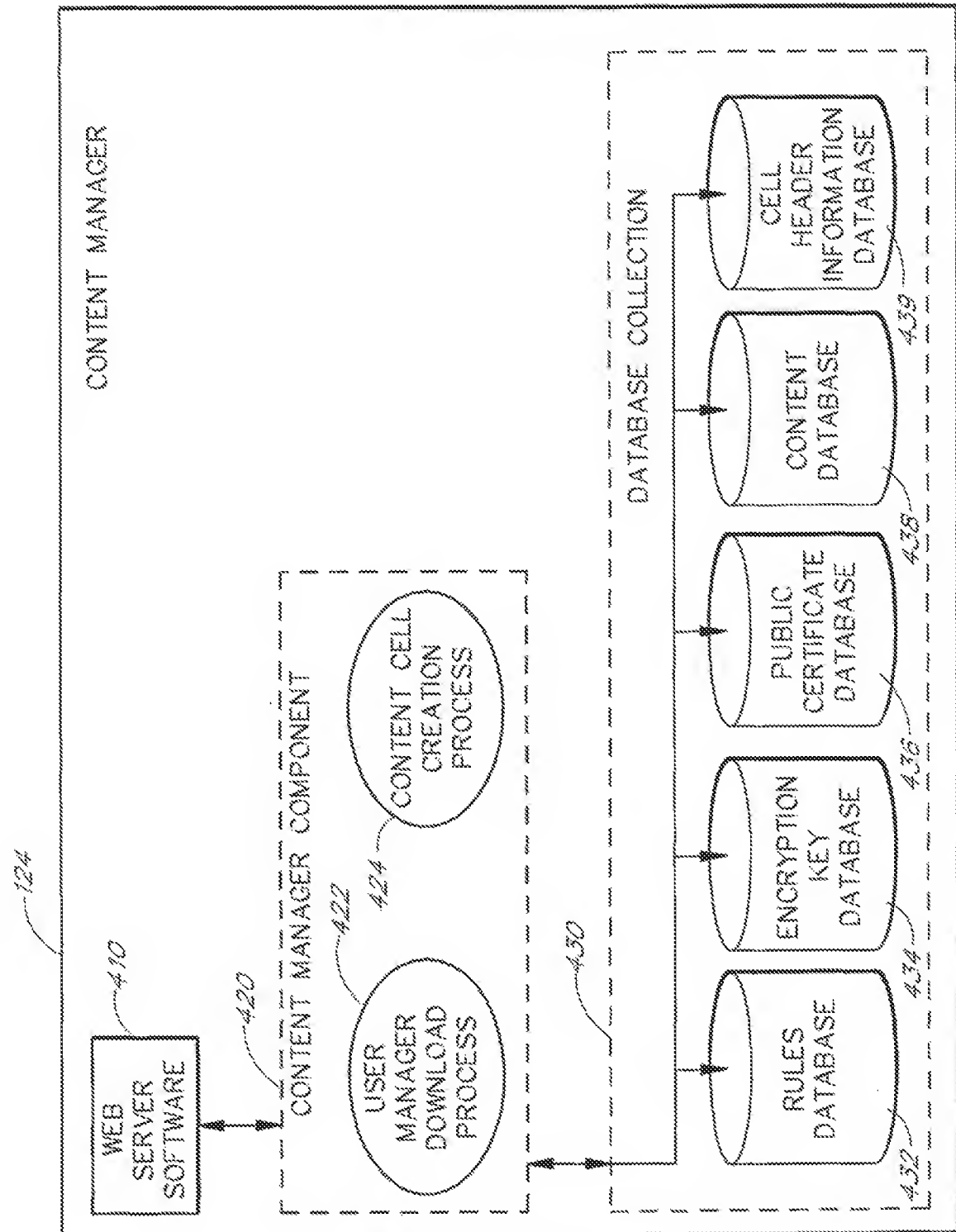


FIG. 3

4/16

FIG. 4



5/16

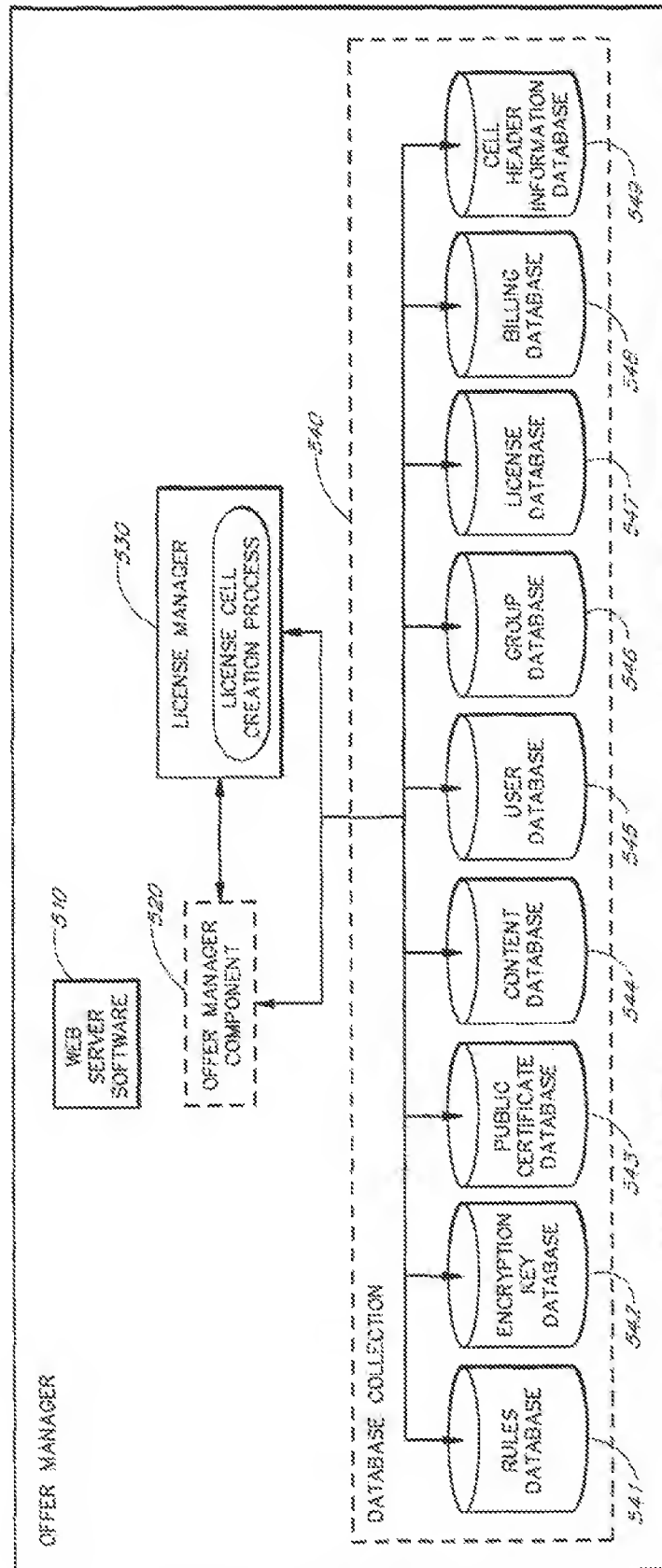


FIG. 5

81/9

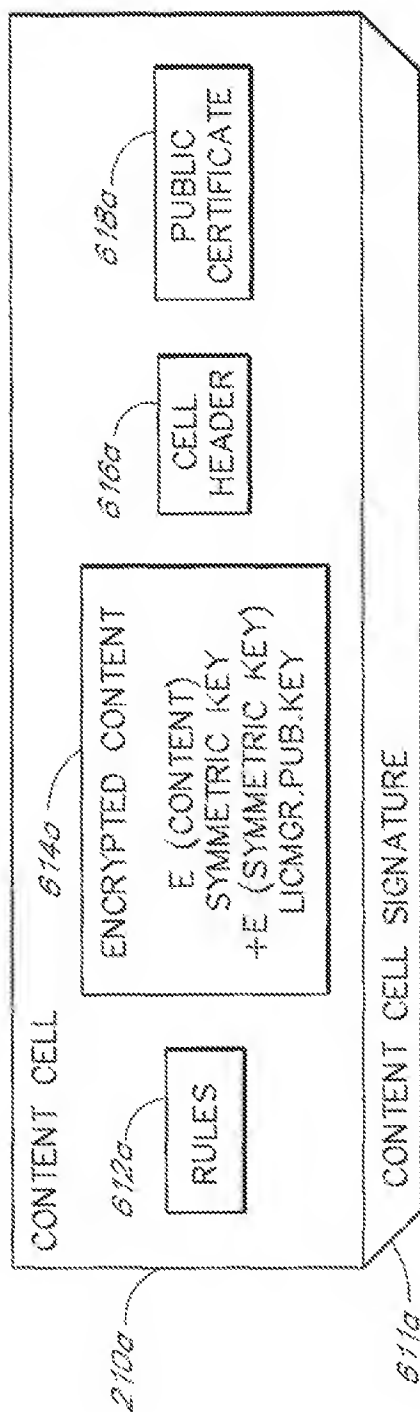


FIG. 6A

7/16

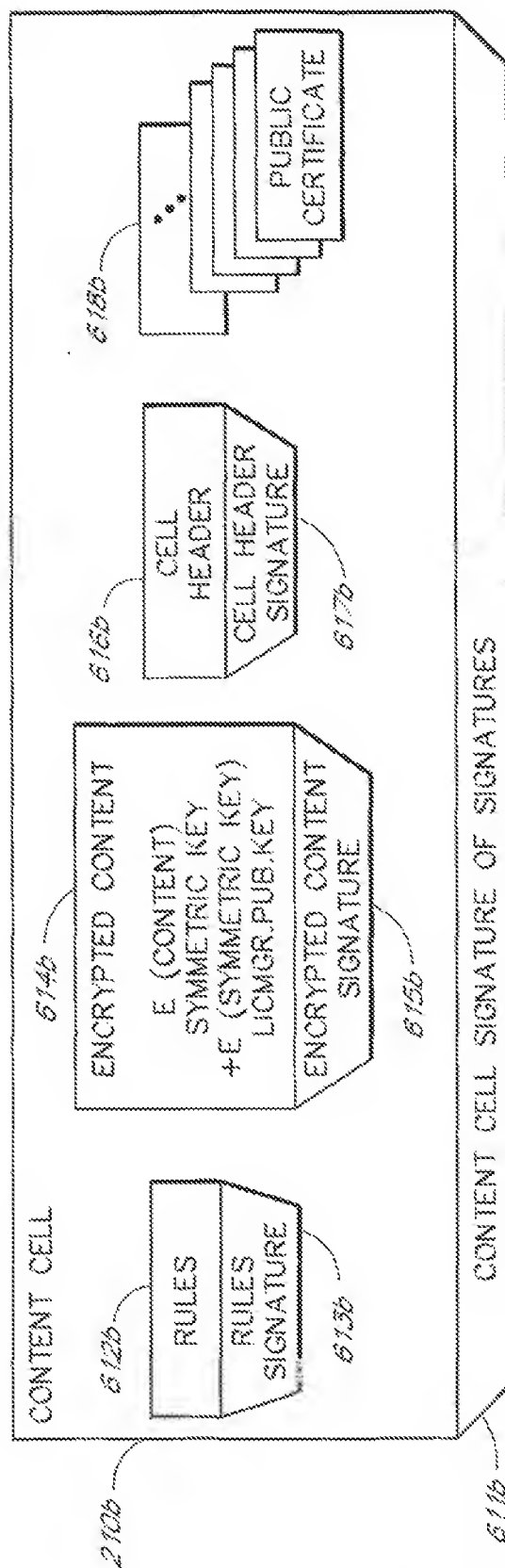


FIG. 6B

8/16

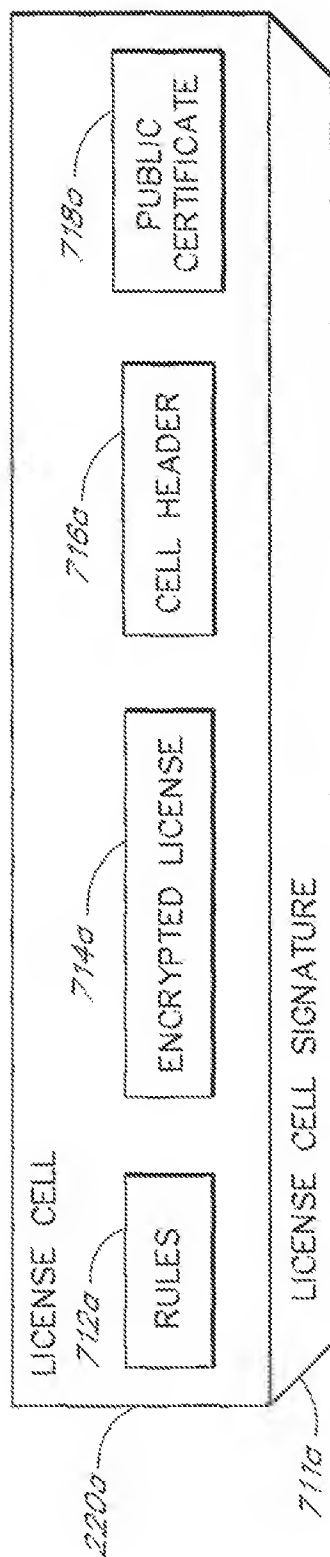


FIG. 7A



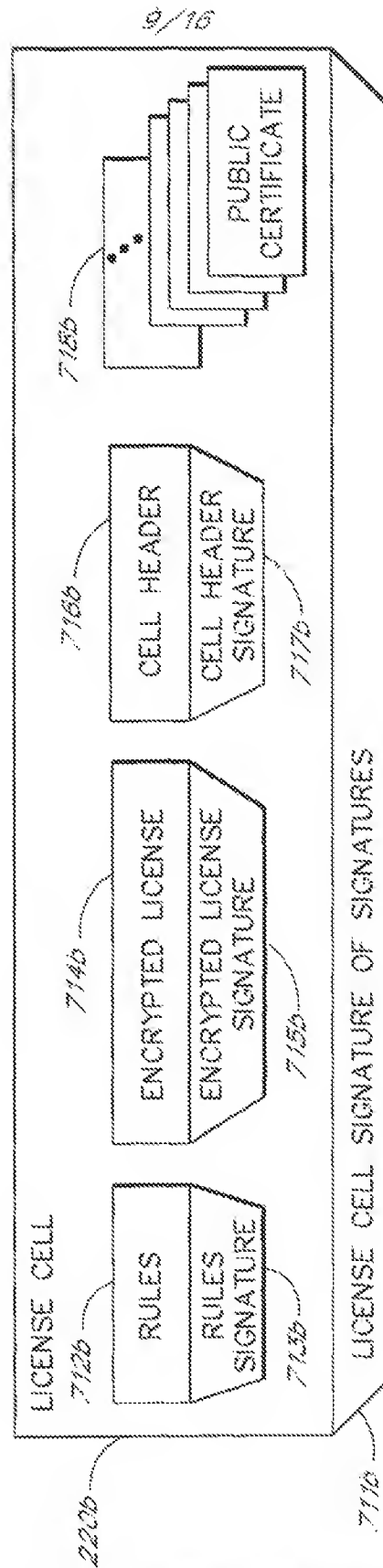


FIG. 7B

10/16

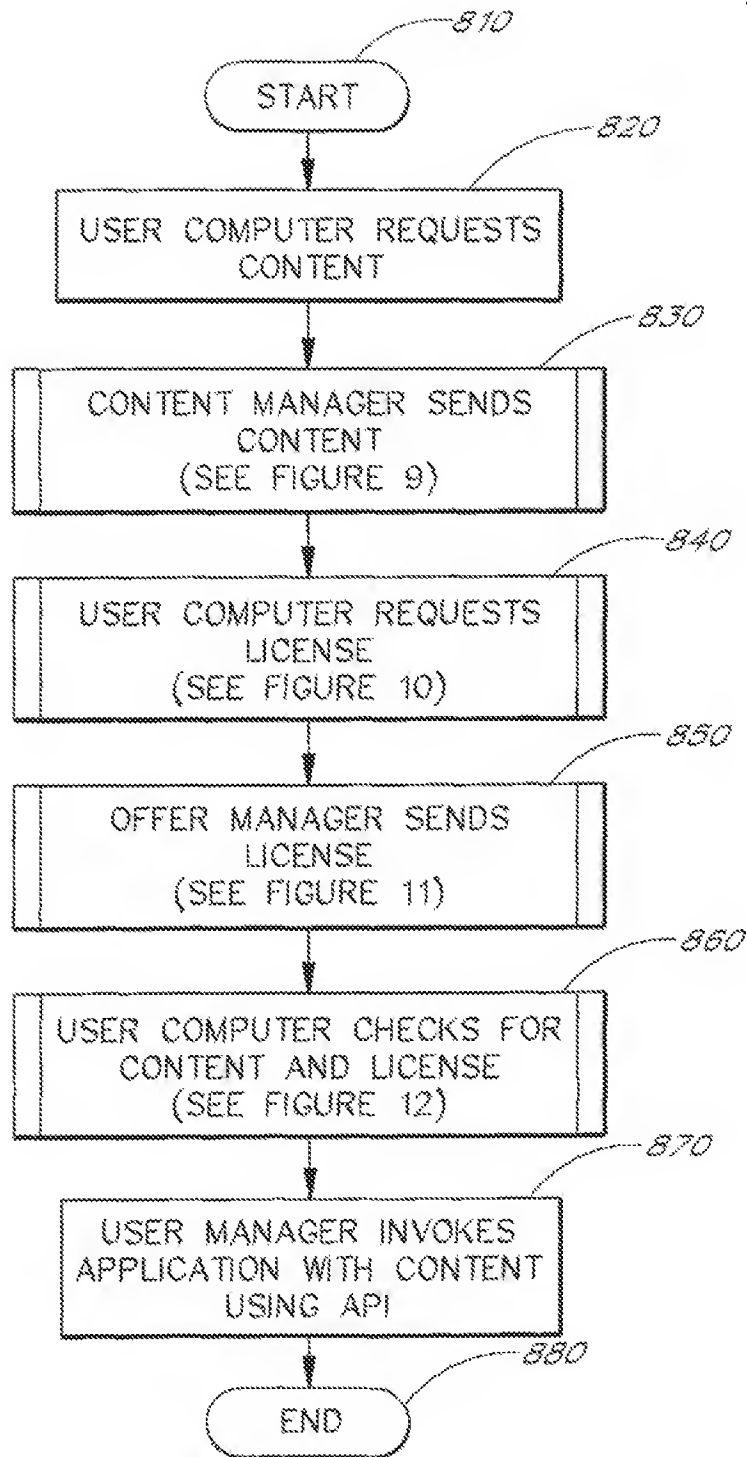


FIG. 8

11/16

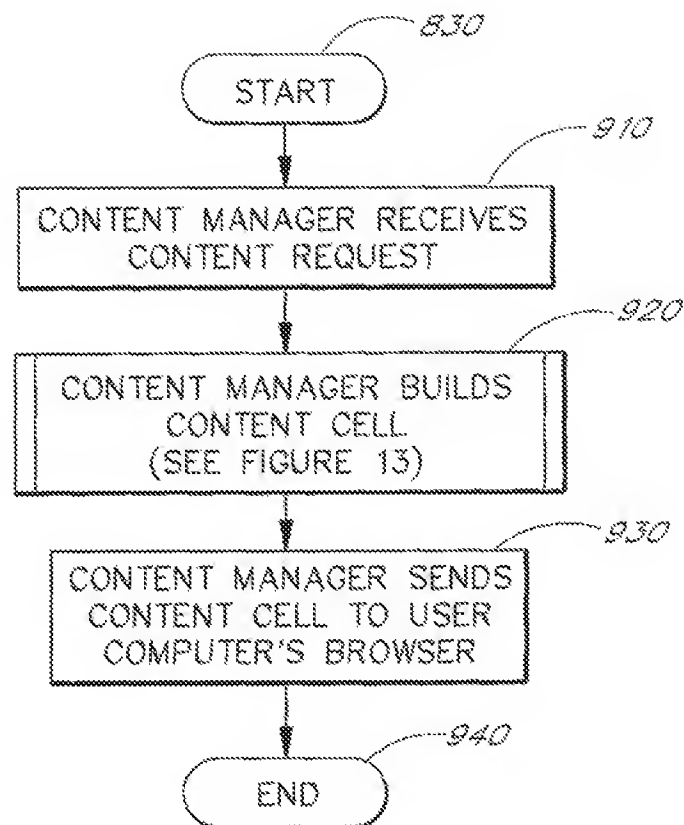


FIG. 9

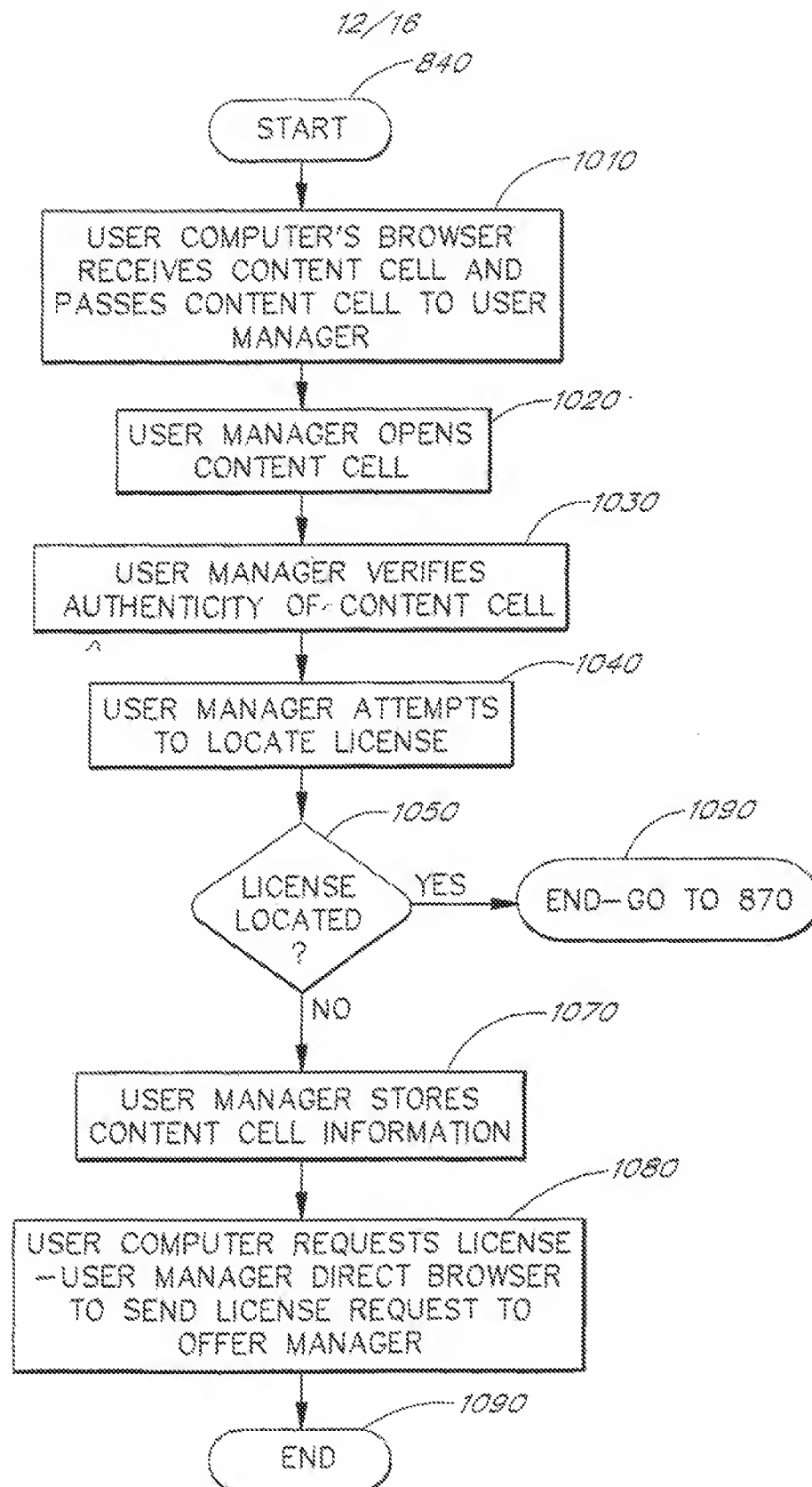


FIG. 10

13/16

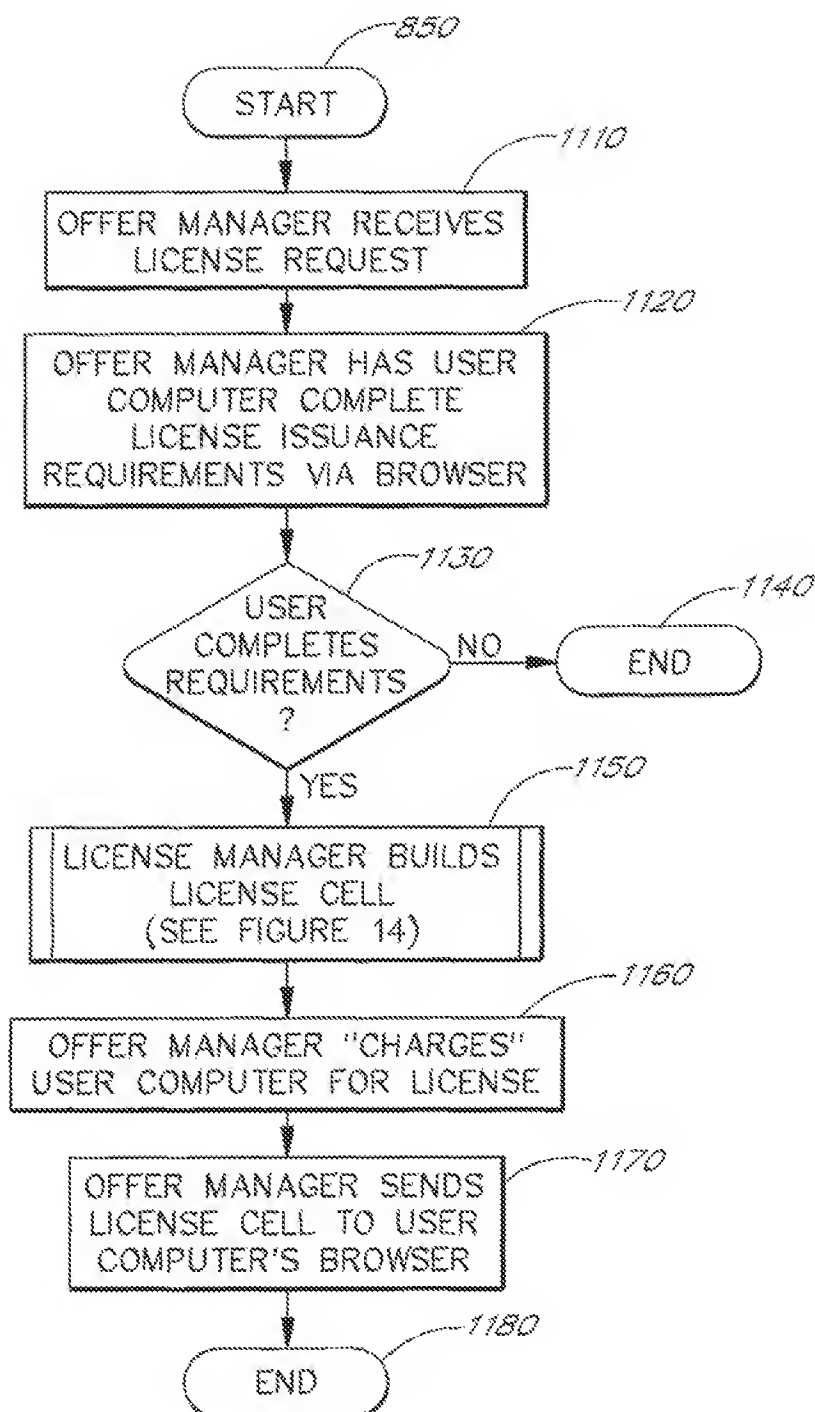


FIG. 11

14/16

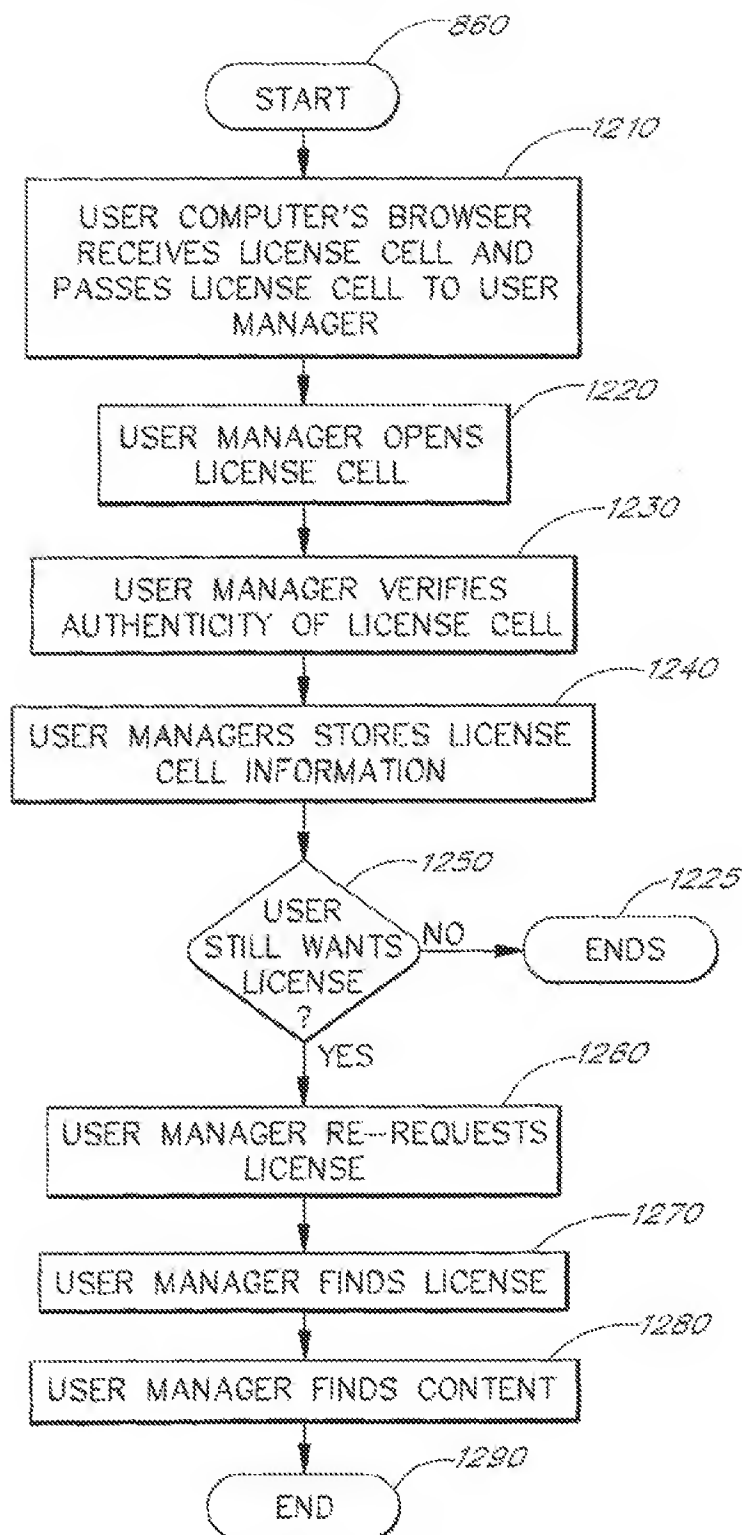


FIG. 12

15/16

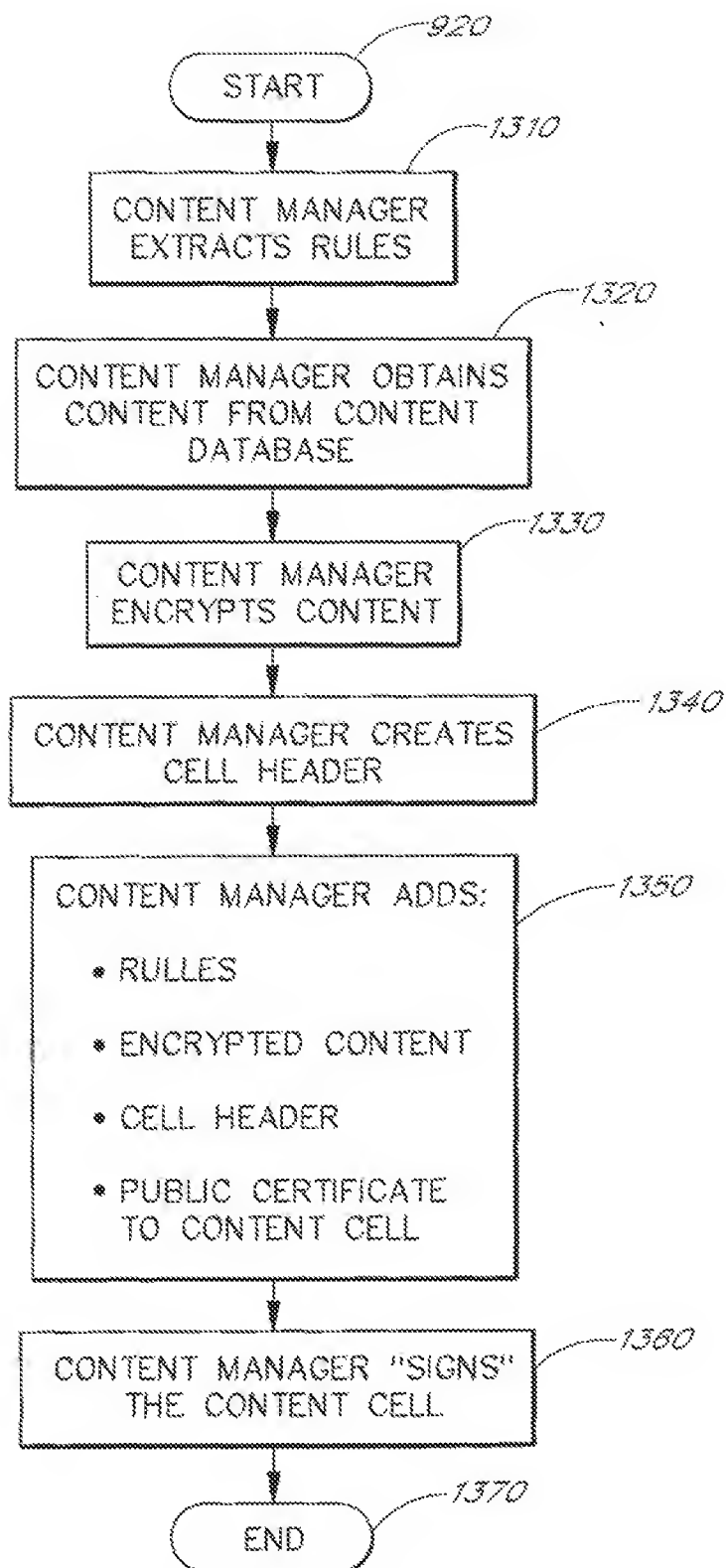


FIG. 13

16/16

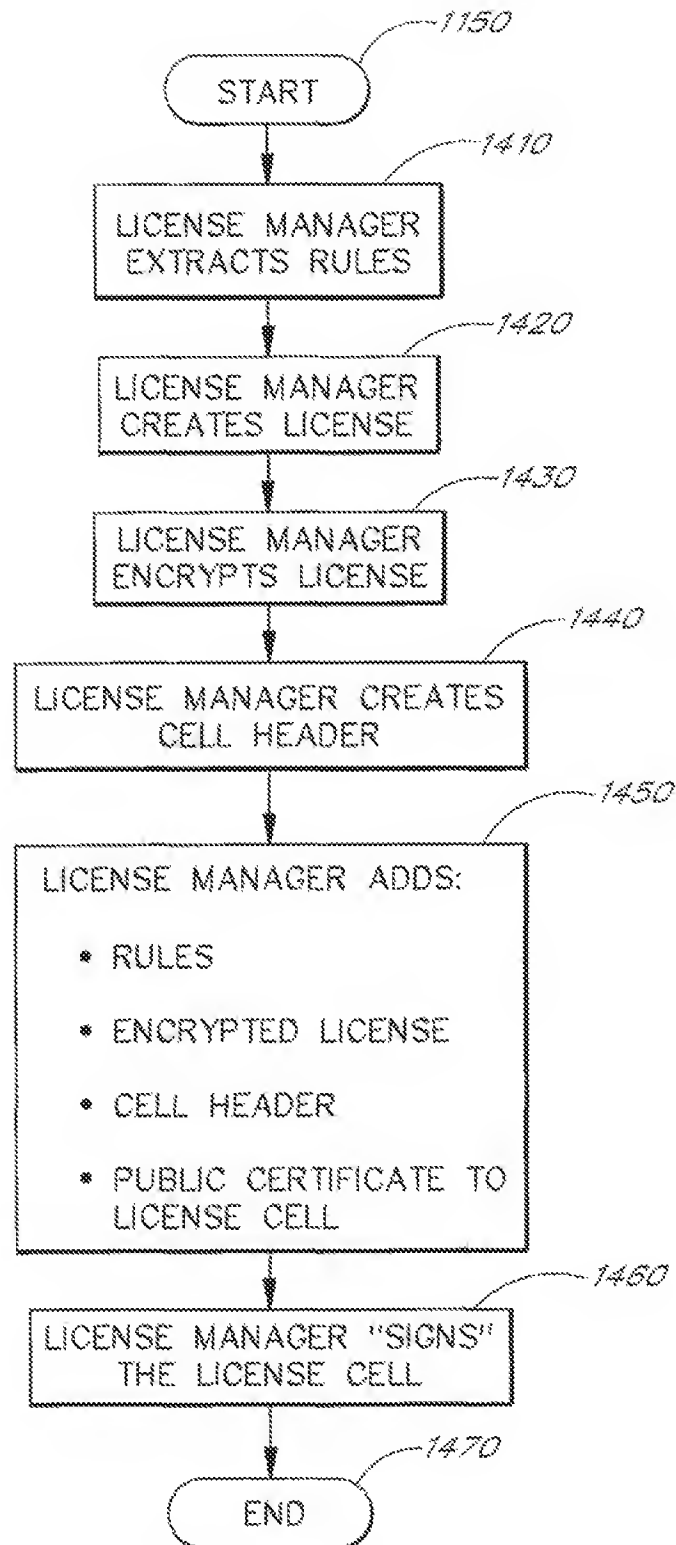


FIG. 14

SUBSTITUTE SHEET (RULE 26)